Teaching a robot to perform task through imitation and on-line feedback

Adrián León, Eduardo F. Morales, Leopoldo Altamirano and Jaime R. Ruiz

National Institute of Astrophysics, Optics and Electronics,
Luis Enrique Erro No. 1, 72840 Tonantzintla, México
{enthe, emorales, robles, jrruiz}@inaoep.mx

Abstract. Service robots are becoming increasingly available and it is expected that they will be part of many human activities in the near future. It is desirable for these robots to adapt themselves to the user's needs, so non-expert users will have to teach them how to perform new tasks in natural ways. In this paper a new teaching by demonstration algorithm is described. It uses a Kinect® sensor to track the movements of a user, eliminating the need of special sensors or environment conditions, it represents the tasks with a relational representation to facilitate the correspondence problem between the user and robot arm and to learn how to perform tasks in a more general description, it uses reinforcement learning to improve over the initial sequences provided by the user, and it incorporates on-line feedback from the user during the learning process creating a novel dynamic reward shaping mechanism to converge faster to an optimal policy. We demonstrate the approach by learning simple manipulation tasks of a robot arm and show its superiority over more traditional reinforcement learning algorithms.

Keywords: robot learning, reinforcement learning, programming by demonstration, reward shaping

1 Introduction

The area of robotics is rapidly changing from controlled industrial environments into dynamic environments with human interaction. To personalize service robots to the user's needs, robots will need to have the capability of acquiring new tasks according to the preferences of the users and non-expert users will have to be able to program new robot tasks in natural and accessible ways. One option is to show the robot the task and to let the robot imitate the user's movements in what is called Programming by Demonstration (PbD) [4]. This approach, however, normally uses sophisticated hardware and can only reproduce the traces provided by the user, so the performance of the robot depends on the performance of the user in the task. An alternative approach is to use reinforcement learning (RL) and let the robot explore the environment to learn the task [12]. This, however, normally results in long training times.

In this paper, the user shows the robot how to perform a task. To capture the user's demonstration, rather than using a sophisticated arrangement of sensors

or special purpose environments, we use a Kinect® sensor to capture the depth information of obstacles and to detect the movements follow by the arm when showing how to perform a particular task. Instead of trying to reproduce exactly the same task, we use reinforcement learning to refine the traces produced by the user. Rather than waiting for the RL algorithm to converge, the user can provide, during the learning process, on-line feedback using voice commands that are translated into additional rewards. We demonstrate the approach in a simple manipulation task.

The rest of the paper is organized as follows. Section 2 reviews the most closely related work. Section 3 describes the proposed method. In Section 4 the experimental set-up is described and the main results presented. Finally Section 5 gives conclusions and future research directions.

2 Background and Related Work

Programing by Demonstration (PbD), Learning from Demonstration (LfD) or Learning by Imitation (LbI), is a mechanism that combines machine learning techniques with human-robot interaction. The idea is to derive control policies of a particular task from traces of tasks performed by a user [3]. One of the advantages of this approach is that the search space is significantly reduced as it is limited to the space used in the demonstration [4]. Several approaches have been used for PbD, however, in most cases the user needs to wear special equipment under particular conditions, limiting its applicability to restricted environments. In this paper, we use a Kinect® sensor which is relatively cheap and robust to changes in illumination conditions. Also, in most of these developments the performance of the system strongly depends on the quality of the user's demonstrations. In this paper, we couple the user's demonstration with a reinforcement learning algorithm to improve over the demonstrations given by the users.

Reinforcement Learning (RL) is a technique used to learn in an autonomous way a control policy in a sequential decision process. The general goal is to learn a control policy that produces the maximum total expected reward for an agent (robot) [12]. Learning an optimal control policy normally requires the exploration of the whole search space and very large training time. Different approaches have been suggested to produce faster convergence times, such as the use of abstractions, hierarchies, function approximation, and more recently reward shaping [11, 9, 10, 1, 8, 5]. In reward shaping, most of these methods require domain knowledge to design an adequate reward shaping function, or try to learn the reward functions with experience, which can take long training times. In our case, the user can provide feedback to the robot and change the reward function. Some authors also have provided feedback from the user and incorporated it into the reinforcement learning algorithm [6, 2, 7]. In [2] the robot first derives a control policy from user's demonstrations and the teacher modifies the policy through a critiquing process. A similar approach is taken in [6], however the user's critique is incorporated into the optimization function used to learn the policy. In [7], the authors combine TAMER, an algorithm that models a hypothetical human reward function, with eight different reward shaping functions.

Contrary to these approaches, in our work the user can provide, through voice commands, feedback that can be given at any time during the learning process and that directly affects the reward function (see also [13]). We extend this last work with traces given by the user and observed by the robot, and with a more powerful representation language to create more general policies, as explained in the next section.

3 Method and System Design

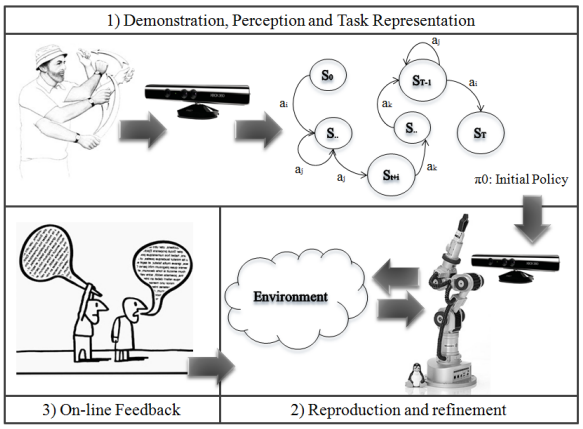

Fig. 1. The imitation and feedback learning

Our approach, illustrated in Figure 1, has three main modules: 1) demonstration, perception and representation of the task, 2) reproduction and refinement, and 3) on-line user feedback.

The interaction between the different components of the system is shown in Figure 2, where the initial demonstrations are used to seed the initial Q-values and the system follows a process where the user can intervene during the RL process.

3.1 Demonstration, perception and task representation

In the demonstrations, the instructor shows the robot the task to learn with his/her arm movements. The 3D positions of the hand and of the objects in the environment are tracked using the Kinect® sensor. These 3D coordinates sequences are obtain from a previously calibrated working area that includes all the arm movements. The sequences are processed to obtain relational state-action pairs. Each state $s \in S$ is described by a six-term tuple with the following elements: $s = (H, W, D, dH, dW, dD)$, where:

- H = Height: $\{Up, Down\}$
- W = Width: $\{Right, Left\}$

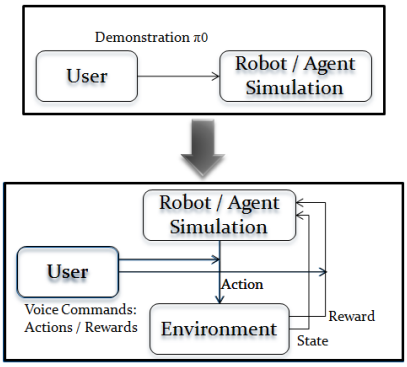

Fig. 2. Training phases of the proposed approach

- D = Depth: $\{Front, Back\}$
- dH = Height distance to target: $\{VeryFar, Far, Close, VeryClose, Over\}$
- dW = Width distance to target: $\{VeryFar, Far, Close, VeryClose, Over\}$
- dD = Depth distance to target: $\{VeryFar, Far, Close, VeryClose, Over\}$

Each action $a \in A$ is described as a movement in one direction with information of how much to move the manipulator, $a = (D, pD)$, where:

- D : Direction $\{Up, Down, Right, Left, Front, Back\}$
- pD : a real value that defines the magnitude of the movement performed by the robot according to how close it is from an object. For example, a *right* movement will have a greater displacement to the right when it is far from the target object than a *right* movement when it is close to the target object.

The main advantage of this representation is that, since it is a relative position between the human or robotic arm with the target place or object, it does not need to have any special transformation between the traces shown by the user and the traces used by the robot. On the other hand, the states and the learned policies, as it will be shown later, are consequently relative to the target object so the initial position of the robot arm and the initial and final position of the target object or place can be completely different from the positions shown by the user, and the learned policy is still suitable for the task.

3.2 Reproduction and refinement

The goal of this stage is to improve over the traces performed by the user. Given a set of initial traces by the user, these are transformed into the state-action pairs with the previously described representation and directly used by the robot to initialize the Q-values of the visited state-action pairs. The robot then follows a normal RL algorithm using Q-learning to improve over the initial policy. During the exploration moves, the robot can reach previously unvisited states that are incrementally added to the state space.

Also, during the execution of actions it is possible to produce continuous actions by combining the discrete actions of the current policy. This is performed as a lineal combination of the discrete actions with the larger Q-values. The lineal combination is proportional to the magnitude of the used Q-values. The updating function over the Q-values is also proportionally performed over all the involved discrete actions.

3.3 On-line feedback

While the robot is exploring the environment to improve over its current policy, the user can provide on-line voice feedback to the robot. We build over the work described in [13], where a fixed vocabulary was defined for the user's commands. The user feedback can be in the form of action commands or as qualifiers over particular states that are transformed into rewards and added to the current reward function.

Our reward function is defined as: $R = R_{RL} + R_{user}$ where R_{RL} is the normal reward function and R_{user} is the reward obtained from the voice commands given by the user. The main difference with previous reward shaping functions is that in our case the rewards can be given sporadically and can be contrary to what it is needed for achieving a goal. Nevertheless, we assume that when they are given correctly they reinforce the movements where the agent is moving towards the goal and satisfy a potential-based shaping framework. So even with noisy feedback from the user we can still guarantee convergence towards an adequate policy as long as the agent receives in average correct rewards (see [13] for more details).

4 Experiments and Results

We used a 6 DOF robot manipulator, named Armonic Arm 6M (see Figure 3 right), in our experiments and the task was to pick-up an object and place it in a new position.

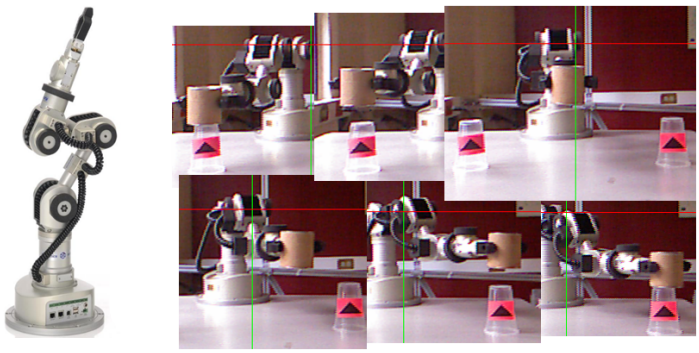

Fig. 3. Robot Katana Armonic Arm 6M

In front of the Kinect sensor, the user simply picks up an object from a spatial position and places it in a different location. The sensor is responsible for identifying the 3D location of the user hand and object and track the hand movements. From the Kinects tracking system we get a sequence of 3D coordinates to define distances and locations with respect to the object and to determine relational states to characterize the task. Figure 4 shows a human demonstration used to pick-up a object and place it in a different location (up) and the information obtained by the Kinect sensor (down).

Figure 3 shows to the left a sequence performed by the robot after learning this task.

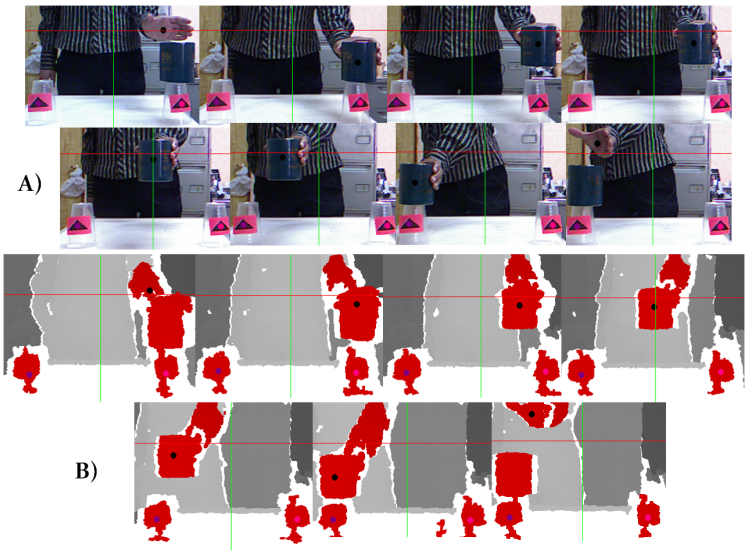

Fig. 4. Human demonstration for picking-up and placing a particular object

For the experiments, we designed different conditions to test the individual parts of the proposed system including a simulator for training during 50 episodes:

1. Using only Reinforcement Learning (RL)
2. Reinforcement Learning + Human demonstration (HD)
3. Reinforcement Learning + Simulation (S) + Human demonstration
4. Reinforcement Learning + Simulation + Human demonstration + User's Feedback (FB)

Figure 5 shows the performance of the different experiments and table 4 shows the total computer times. As can be seen, using human demonstration and user's feedback during the learning process can significantly reduced the convergence times for the RL algorithm. It should be noted that each episode

shown in the figure started from random initial positions and ended in random (reachable) object positions.

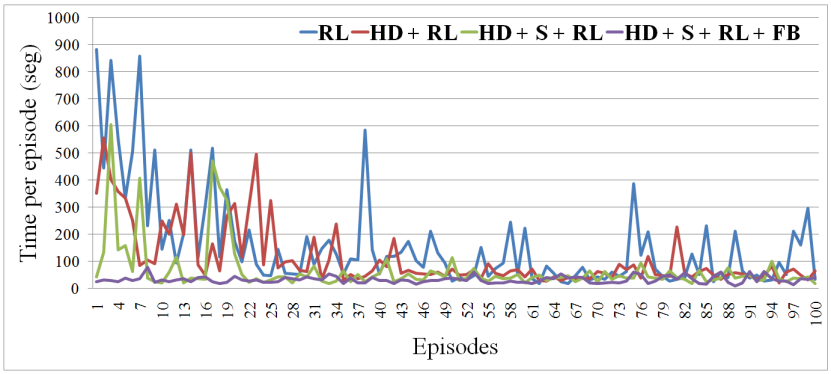

Fig. 5. Performance of the different experimental conditions. (i) RL = reinforcement learning, (ii) $HD + RL$ = RL + human demonstration, (iii) $HD + S + RL$ = RL + simulation traces + human demonstrations, and (iv) $HD + S + RL + FB$ = RL + simulation traces + human demonstrations + user's feedback.

Table 1. Total computing times: The second row shows the time of HD (˜5 min) and the time of RL. The third and fourth rows show the time of HD, S, and RL respectively in each column; FB does not require additional time. The last column shows the total time spent for each experimental condition.

	Time (s)			Total time (s)
RL			16168.896	16168.896
HD + RL	˜300		11056.56	11356.56
HD + S + RL	˜300	25.628	6729.399	7055.027
HD + S + RL + FB	˜300	19.348	3242.273	3561.621

5 Conclusions and Future Work

Teaching a robot how to perform new tasks will soon become a very relevant topic with the advent of service robots. We want non-expert users to be able to teach robots in natural ways how to perform a new task. In this paper, we have described how to teach a robot to perform a task by combining demonstration performed by the user with voice feedback over the performance of the robot during its learning phase. Our main contributions are: the simple PbD setup with Kinect sensor, the representation used for the demonstration which is used

also in RL and the incorporation of on-line voice feedback from the user during the learning process.

There are several research directions that we would like to pursue. So far we have focused our approach in the displacement of the hand and of the end effector. This is suitable in environment; without obstacles or in static environments. As future work, we would like to incorporate information from the movements of all the articulations. We would also like to enrich the vocabulary for other stages in the learning process, like assigning particular names to learned sub-tasks and then re-using them for learning more complex tasks. Finally we would like to test our approach in other maniplation tasks and with different objects.

References

1. Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *In Proceedings of the Twenty-first International Conference on Machine Learning*. ACM Press, 2004.
2. Brenna Argall, Brett Browning, and Manuela Veloso. Learning by demonstration with critique from a human teacher. In *2nd Conf. on Human-Robot Interaction (HRI)*, pages 57–64, 2007.
3. Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration, 2009.
4. Aude G. Billard, Sylvain Calinon, Ruediger Dillmann, and Sstefan Schaal. *Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics*, chapter 59. Springer, New York, NY, USA, 2008.
5. Marek Grzes and Daniel Kudenko. Learning shaping rewards in model-based reinforcement learning, 2009.
6. Kshitij Judah, Saikat Roy, Alan Fern, and Thomas G. Dietterich. Reinforcement learning via practice and critique advice. In *AAAI*, 2010.
7. W. Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning, 2010.
8. George Konidaris and Andrew Barto. Autonomous shaping: knowledge transfer in reinforcement learning. In *In Proceedings of the 23rd Internation Conference on Machine Learning*, pages 489–496, 2006.
9. A. Laud. Theory and application of reward shaping in reinforcement learning, 2004.
10. Maja J Mataric. Reward functions for accelerated learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 181–189. Morgan Kaufmann, 1994.
11. Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287. Morgan Kaufmann, 1999.
12. Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. The MIT Pres, Cambridge, MA, London, England, 1998.
13. Ana C. Tenorio-Gonzalez, Eduardo F. Morales, and Luis Villaseñor Pineda. Dynamic reward shaping: training a robot by voice. In *Proceedings of the 12th Ibero-American conference on Advances in artificial intelligence*, IBERAMIA'10, pages 483–492, Berlin, Heidelberg, 2010. Springer-Verlag.