Teaching a Robot New Tasks through Imitation and Feedback

Adrián León ENTHE@INAOEP.MX
Eduardo F. Morales EMORALES@INAOEP.MX
Leopoldo Altamirano ROBLES@INAOEP.MX
Jaime R. Ruiz JRRUIZ@INAOEP.MX

National Institute of Astrophysics, Optics and
Electronics, Luis Enrique Erro 1, Sta. Ma.
Tonantzintla, Puebla 72840 Mexico

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 a natural way. In this paper a new teaching by demonstration algorithm is described. It uses a Kinect® sensor to track the movements of a user, it represents the tasks with a relational representation to facilitate the correspondence problem between the user and robot arm and to learn a more general policy, 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.

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 acquire new tasks according to the preferences of the users, so non-expert users will have to be able to program new robot tasks in natural and accessible ways. Several approaches have been proposed for learning how to perform tasks in robotics, but perhaps the most widely used have been Programming by Demonstration (PbD) and Reinforcement Learning (RL). In Programming by Demonstration the task to learn is shown to the robot and the goal for the robot is to imitate the demonstration to complete the task (Billard et al., 2008). PbD combines machine learning techniques with human-robot interaction and the idea is to derive control policies of a particular task from traces of tasks performed by a teacher (Argall et al., 2009). 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 (Billard et al., 2008).

Several approaches have been proposed in PbD, however, in most cases the user needs to wear special equipment under particular conditions (Ijspeert et al., 2002; Aleotti & Caselli, 2007; Calinon & Billard, 2007), limiting its applicability to restricted environments. In this paper, 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. The Kinect® sensor is relatively cheap, it is robust to changes in illumination conditions, and it is not attached to the user. Also, in most of the research work in PbD the performance of the system strongly depends on the quality of the user's demonstrations. In this paper, instead of trying to reproduce exactly the same task, we use reinforcement learning to refine the traces produced by the user.

Reinforcement Learning is another popular approach that has been used in robotics to learn how to perform a task. RL can be characterized as an MDP

Appearing in *Proceedings of the ICML Workshop on New Developments in Imitation Learning*, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s).

$< S, A, R, P >$ where: S is a set of state, A is a set of actions, R is a reward function, and P a probability state transition function. The general goal is to learn a control policy that produces the maximum total expected reward for an agent (robot) (Sutton & Barto, 1998). Learning an optimal control policy normally requires the exploration of the whole search space and very large training time and different approaches have been suggested to ameliorate this, such as the use of abstractions, hierarchies, function approximation, and more recently reward shaping (Ng et al., 1999; Laud, 2004; Mataric, 1994; Abbeel & Ng, 2004; Konidaris & Barto, 2006; Grzes & Kudenko, 2009). In reward shaping, most of the previous work define a fixed reward shaping function that is used during the whole learning process. In this paper, rather than waiting for RL to converge to an optimal policy, the robot tries to perform the task with its current (suboptimal) policy and, the user can provide on-line feedback using voice commands that are translated into additional rewards. This creates a novel dynamic reward shaping approach that can be used to accelerate the learning process and to correct the initial traces. We demonstrate the approach in a simple manipulation task.

The remainder of the paper is structured 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. Related work

Several approaches have been proposed in PbD. In (Ijspeert et al., 2002), a Locally Weighted Regression (LWR) approach is used to show how to perform tasks to an anthropomorphic robot. The tasks are based on desired trajectories that can be imitated using kinematics variables. The system described can combine several trajectories; re-use trajectories previously learned and deal easily with the correspondence problem.

A similar work is presented in (Calinon & Billard, 2007), where a fujitsu HOAP-3 humanoid robot can learn basketball referee signals from human demonstration. It uses Gaussian Mixture Regression (GMM) to reconstruct the shown task. The approach combines programming by human demonstration and kinesthetic teaching that allows naturally looking trajectories and also tackle the correspondence problem as the previous work.

Some advantages of these works are that a human teacher is involved to demonstrate the task to a robot with her/his own body, and can significantly reduce the correspondence problem. However, in both cases special wearable sensors are needed to detect and record human movements which limit their applicability to restricted environments. Also in most of the PbD research, the performance of the system strongly depends on the quality of the user's demonstrations.

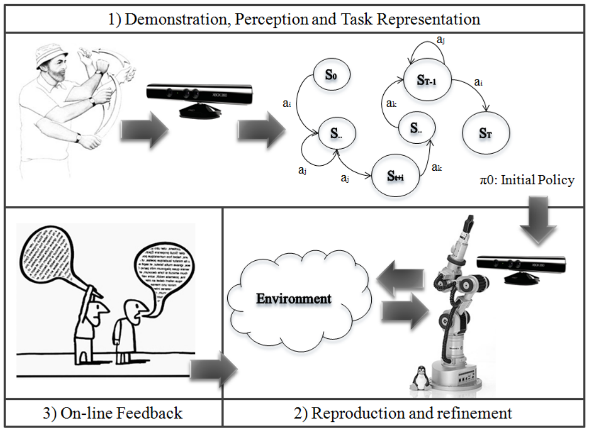

Figure 1. The imitation and feedback learning

Some authors have provided feedback from the user and incorporated it into the reinforcement learning algorithm (Judah et al., 2010; Argall et al., 2007; Knox & Stone, 2010). In (Argall et al., 2007) 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 (Judah et al., 2010), however the user's critique is incorporated into the optimization function used to learn the policy. In (Knox & Stone, 2010), the authors combine TAMER, an algorithm that models a hypothetical human reward function, with eight different reward shaping functions. In (Tenorio-Gonzalez et al., 2010), the user provides, through voice commands, feedback that can be given at any time during the learning process acting as a dynamic reward shaping function. In this paper, this last work is extended by incorporating demonstrations from the user observed by the robot and by using a more powerful representation language to create more general policies.

3. Learning from Human Demonstration and Feedback

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.

In the stage of 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. The sequences are processed to obtain for each frame relational state-action pairs. Each state $s \in S$ is a six-term tuple describing the 3D position and distance of the hand or end effector with respect to the target object or target place.

$$s = (H, W, D, dH, dW, dD) \text{ where:}$$

Position of hand or end effector with respect to the target object or position:

- H (Height) = {*Up, Down*}
- W (Width) = {*Right, Left*}
- D (Depth) = {*Front, Back*}

Distance of the hand or end effector to the object or target position:

- dH (Height) = {*VeryFar, Far, Close, VeryClose, Over*}
- dW (Width) = {*VeryFar, Far, Close, VeryClose, Over*}
- dD (Depth) = {*VeryFar, Far, Close, VeryClose, Over*}

Each action $a \in A$ from the sequence is described as a rotational movement in each direction ((*Up, Null, Down*), (*Right, Null, Left*), (*Front, Null, Back*)).

The main advantage of this representation is that, since it is a relative distance between the human or robotic arm with the target object or position, it does not need to have any special transformation between the traces shown by the user and the traces used by the robot. Furthermore, the initial position of the end effector or hand and the initial and final positions of the target object or target position can be completely different from the positions shown by the user in the traces, and the learned policy is still suitable for the task.

The traces of the task to learn given by the user are translated into states and actions with the previously defined values. Once a transformed set of state-action pairs is obtained, it is used to initialize the Q-values of the state-action pairs involved in the provided traces. 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. The actions are associated with information of how much to move the manipulator depending on how close its target position. For example, a *Right* move has a greater displacement to the right when it is far from the target object than a *Right* move when it is close to the target position.

Table 1. Rewards given by the user during the learning process.

Word	Value of Reward
Objetivo (Goal)	100
Excelente (Excellent)	10
Bien (Good)	5
Terrible (Terrible)	-10
Mal (Bad)	-5

Table 2. Actions given by the user during the learning process

Words (Actions)	
Arriba (Up)	Abajo (Down)
Derecha (Right)	Izquierda (Left)
Frente (Front)	Atrás (Back)

The robot initially moves one DOF at a time, it is possible to produce combined actions by producing lineal combinations of the discrete action with larger Q-values. The algorithm takes the three actions with larger Q-values and generates a combined action which is the resulting vector of those actions. The updating function over the Q-values, in this case, is also proportionally performed over all the involved discrete actions. So although the action's space is discrete, this simple mechanism produces smoother movements involving more than one action at the same time.

While the robot is learning, the user can provide on-line voice feedback to the robot. We build over the work described in (Tenorio-Gonzalez et al., 2010), 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. The provided feedback is translated into rewards values that are added to the current reward function. Tables 1 and 2 show examples of qualifiers and their associated rewards and the actions that the user can provide to change the behavior of the robot.

So our reward function is defined as: $R = R_{RL} + R_{user}$, where R_{RL} is the traditionally defined reward shaping 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

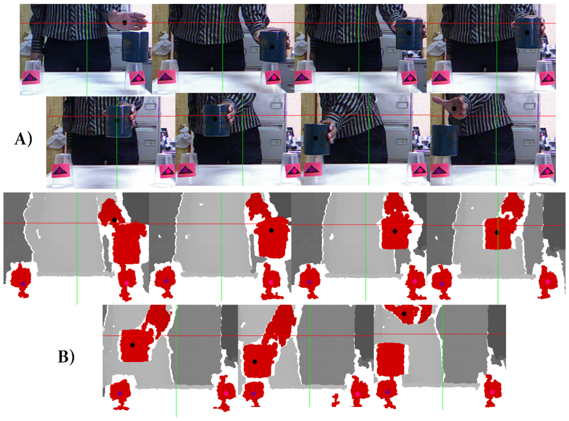

Figure 2. Human demonstration for picking-up and placing a particular object

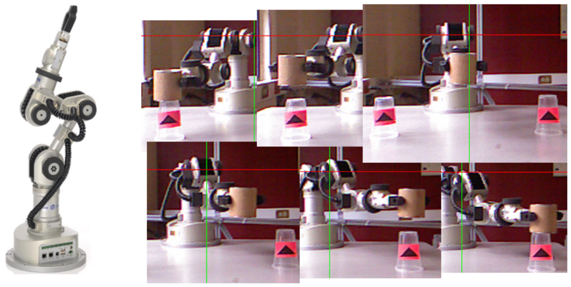

Figure 3. Robot Katana Armonic Arm 6M

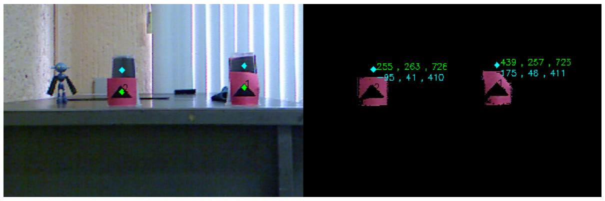

Figure 4. Locations using color-based segmentation algorithm

in our case additional 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 when 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 (Tenorio-Gonzalez et al., 2010) for more details).

It is also possible to interleave the use of a simulator to accelerate the learning process between the trials of the robot. In the following section we show results with a robotic arm and with a simulator.

4. Experiments and Results

To prove our approach, in our experiments, we used a 6 DoF robot manipulator called Armonic Arm 6M (see Figure 3 right). The robot simply has to learn to pick-up an object from one position and place it in a new position.

Figure 2 shows a human demonstration used to pick-up an object and place it in a different location (up) and the information obtained by the Kinect sensor (down). To identify the locations where the object is picked and placed, we used a simple color-based segmentation algorithm with the RGB Kinect's camera (Fig 4). Figure 3 shows to the left a sequence performed by the robot after learning this task.

In our learning framework we are incorporating several elements that we believe can help to teach a robot how to perform a new task in a more natural way and converge faster to an adequate policy. In particular, our framework does not depend on costly equipment, wearables or special lighting conditions, and uses a natural teaching setting with the user showing the task, in a possibly inaccurate way, and providing voice feedback during the learning process.

We performed several experiments to test different setting, incorporating human demonstrations, user's feedback and simulated traces (20 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)

We also performed experiments with a simulator, either learning completely the optimal policy with traditional RL or interleaving simulation traces with our learning framework.

Each experiment was repeated 20 times and the tables show the averages of the training times and accumulated rewards. Figures 5, 6, 7, and 8, show the performance time of each setting and Figure 9 shows a graph will all the settings together. Similarly, Figures 10, 11, 12, and 13 show the performance of the different settings and Figure 14 shows all the results in the same figure.

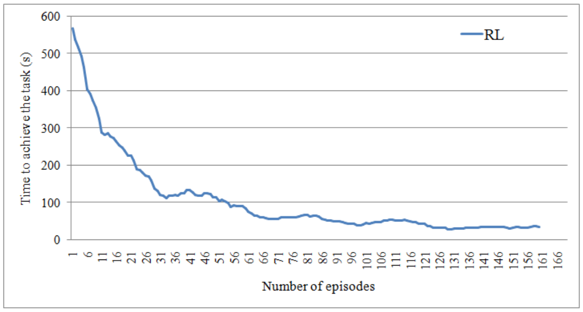

Figure 5. (i) RL = Reinforcement Learning.

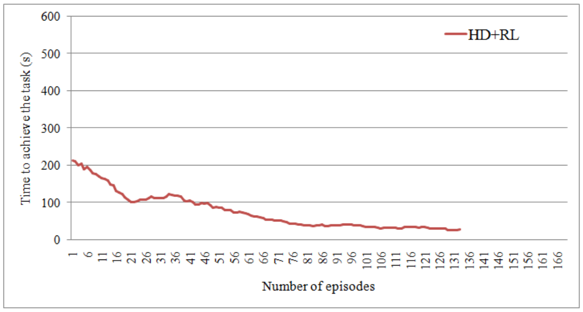

Figure 6. (ii) $HD + RL$ = Human Demonstration + RL.

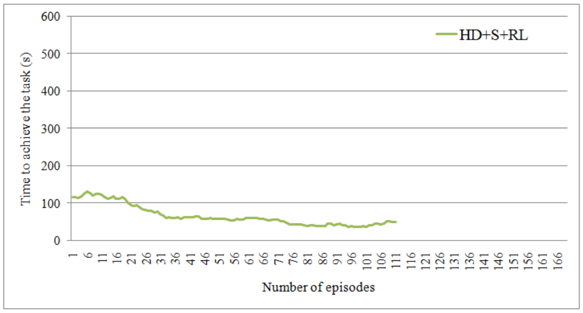

Figure 7. (iii) $HD + S + RL$ = Human Demonstrations + Simulation traces + RL.

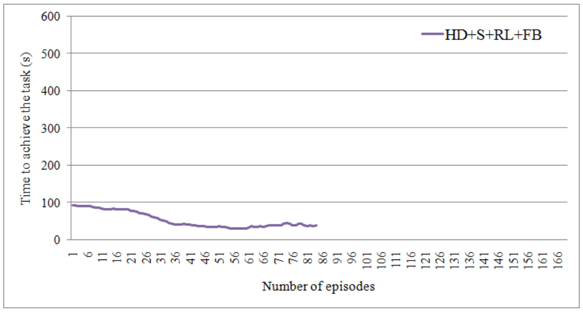

Figure 8. (iv) $HD + S + RL + FB$ = Human Demonstrations + Simulation traces + RL + user's Feedback.

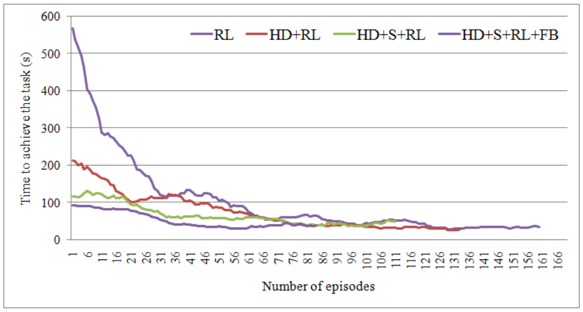

Figure 9. Convergence time for each experimental condition.

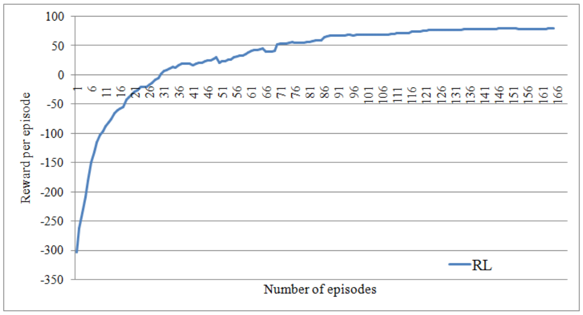

Figure 10. (i) RL = Reinforcement Learning.

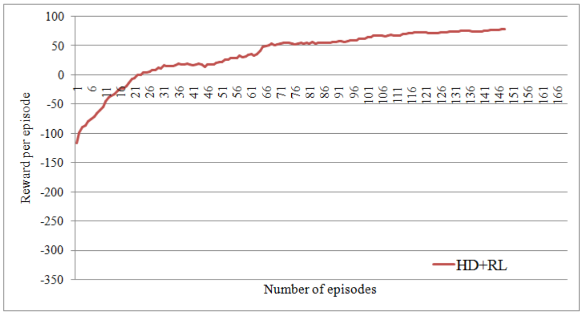

Figure 11. (ii) $HD + RL$ = Human Demonstration + RL.

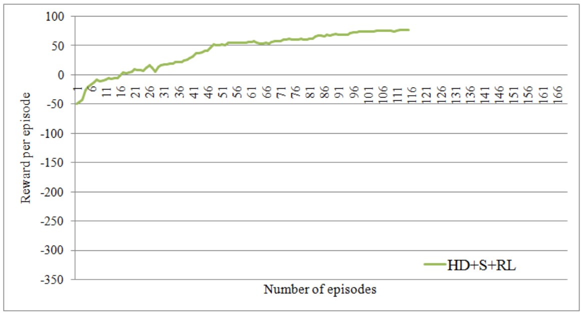

Figure 12. (iii) $HD + S + RL$ = Human Demonstrations + Simulation traces + RL.

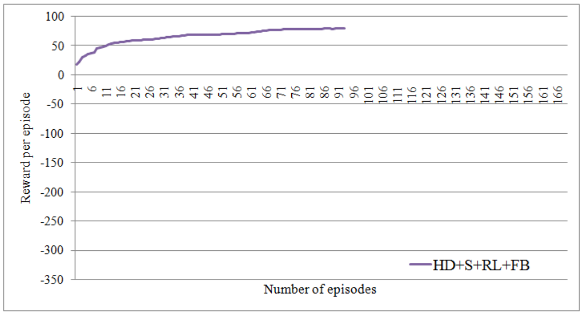

Figure 13. (iv) $HD + S + RL + FB$ = Human Demonstrations + Simulation traces + RL + user's Feedback.

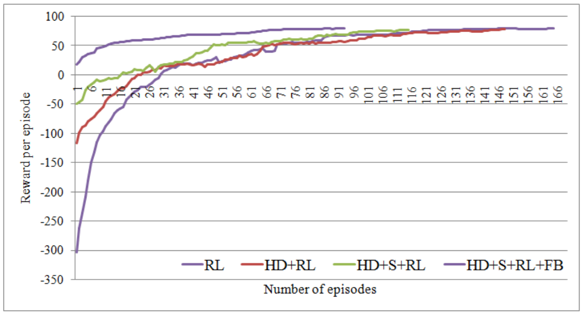

Figure 14. Average of accumulated reward during the training for each of the experimental condition proposed.

As can be seen, using human demonstration and user's feedback during the learning process can significantly reduce the convergence times for the RL algorithm. Table 4 shows the total computer time including the demonstration and simulation time required respectively. It should be noted that each episode shown in the figure started from random initial positions and ended in random (reachable) object positions.

Table 3. Total computing times

Time (s)			Total time (s)
HD	S	RL	
		21799.17	21799.17
˜180		12047.58	12227.58
˜180	3.26	9284.85	9468.11
˜180	4.15	5296.46	5480.61

The first row describes the time used purely with RL, the second one shows the time of human demonstrations (HD) (close to 3 min.) and the time of RL. The third and fourth rows show the time of HD, S, and RL respectively in each column. The last column shows the total time required for each experimental condition.

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 how to perform a new task in natural ways. 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. The main contributions of the approach are the simple PbD set-up with a Kinect sensor, the representation used for the demonstrations, and the incorporation of on-line voice feedback from the user during the learning process that serves as a dynamic reward shaping function and helps to cponverge faster to a suitable policy.

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 and we would like to incorporate information from the movements of all the articulations. We would also like to enrich the vocabulary for a more natural interaction and test the approach in other manipulation tasks with different objects.

References

Abbeel, Pieter and Ng, Andrew Y. Apprenticeship learning via inverse reinforcement learning. In *In Proceedings of the Twenty-first International Conference on Machine Learning*. ACM Press, 2004.

Aleotti, J. and Caselli, S. Robust trajectory learning and approximation for robot programming by demonstration. *in: The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems*, 54(5):409–413, 2007.

Argall, Brenna, Browning, Brett, and Veloso, Manuela. Learning by demonstration with critique from a human teacher. In *2nd Conf. on Human-Robot Interaction (HRI)*, pp. 57–64, 2007.

Argall, Brenna D., Chernova, Sonia, Veloso, Manuela, and Browning, Brett. A survey of robot learning from demonstration, 2009.

Billard, Aude G., Calinon, Sylvain, Dillmann, Ruediger, and Schaal, Sstefan. *Robot programming by demonstration, in: B. Siciliano, O. Khatib (Eds.), Handbook of Robotics*, chapter 59. Springer, New York, NY, USA, 2008.

Calinon, Sylvain and Billard, Aude G. Incremental learning of gestures by imitation in a humanoid robot. *in: Proceedings of the 2nd ACM/IEEE Inter-*

national Conference on Human-Robot Interactions, HRI'07, 2007.

Grzes, Marek and Kudenko, Daniel. Learning shaping rewards in model-based reinforcement learning, 2009.

Ijspeert, Auke Jan, Nakanishi, Jun, and Schaal, Stefan. Movement imitation with nonlinear dynamical systems in humanoid robots. In *In IEEE International Conference on Robotics and Automation (ICRA2002)*, pp. 1398–1403, 2002.

Judah, Kshitij, Roy, Saikat, Fern, Alan, and Dietterich, Thomas G. Reinforcement learning via practice and critique advice. In *AAAI*, 2010.

Knox, W. Bradley and Stone, Peter. Combining manual feedback with subsequent mdp reward signals for reinforcement learning, 2010.

Konidaris, George and Barto, Andrew. Autonomous shaping: knowledge transfer in reinforcement learning. In *In Proceedings of the 23rd Internation Conference on Machine Learning*, pp. 489–496, 2006.

Laud, A. Theory and application of reward shaping in reinforcement learning, 2004.

Mataric, Maja J. Reward functions for accelerated learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pp. 181–189. Morgan Kaufmann, 1994.

Ng, Andrew Y., Harada, Daishi, and Russell, Stuart. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999.

Sutton, Richard S. and Barto, Andrew G. *Reinforcement learning: An introduction.* The MIT Pres, Cambridge, MA, London, England, 1998.

Tenorio-Gonzalez, Ana C., Morales, Eduardo F., and Villaseñor Pineda, Luis. Dynamic reward shaping: training a robot by voice. In *Proc. of the 12th Ibero-American conference on Advances in artificial intelligence*, IBERAMIA'10, 2010.