# Dynamic Reward Shaping: Training a Robot by Voice

Ana C. Tenorio-Gonzalez[⋆], Eduardo F. Morales, and Luis Villaseñor-Pineda

National Institute of Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro #1, 72840 Tonantzintla, México
{catanace17,emorales,villasen}@inaoep.mx
http://www.inaoep.mx

**Abstract.** Reinforcement Learning is commonly used for learning tasks in robotics, however, traditional algorithms can take very long training times. Reward shaping has been recently used to provide domain knowledge with extra rewards to converge faster. The reward shaping functions are normally defined in advance by the user and are static. This paper introduces a dynamic reward shaping approach, in which these extra rewards are not consistently given, can vary with time and may sometimes be contrary to what is needed for achieving a goal. In the experiments, a user provides verbal feedback while a robot is performing a task which is translated into additional rewards. It is shown that we can still guarantee convergence as long as most of the shaping rewards given per state are consistent with the goals and that even with fairly noisy interaction the system can still produce faster convergence times than traditional reinforcement learning techniques.

## 1 Introduction

Service robots are beginning to be used in a wide range of applications such as entertainment, assistance, maintenance, cleanse, transport, and guidance, and it is expected to be commonly found in houses and offices in the near future. A key element for their complete incorporation will be their ability to learn new tasks. Reinforcement Learning (*RL*) [21] has been widely used and suggested as a good candidate for learning tasks in robotics, e.g., [2,20,16,23]. This is mainly because it allows an agent, i.e., the robot, to "autonomously" develop a control policy for performing a new task while interacting with its environment.

The use of reinforcement learning in robotics can be problematic due to large state and action spaces which normally results in long training times. Different approaches have been suggested to produce faster convergence times, such as the use of abstractions, hierarchies, function approximation, and more recently reward shaping.

An alternative approach to teach a robot how to perform a task is with programming by demonstration, in which the user shows a robot an action which is

[⋆] Master's thesis scholarship No. 214262 and research project No. 84162 from Conacyt.

later executed by the robot. This approach, however, normally uses sophisticated hardware and can only reproduce the traces provided by the user.

An alternative, and we believe more natural, approach to make a robot learn a new task, is by telling the robot how to perform it and provide feedback while the robot is executing the task until we are satisfied with its performance. In this paper, the user's feedback is associated with a reward shaping function and is used to produce faster convergence times. This approach, however, pose several problems:

1. Even with simple voice commands, speech recognition systems are not completely reliable which can produce a noisy reward shaping function, so we would like to know under which conditions it is possible to recover from such errors.
2. The user can provide his rewards with certain delay, so we would like to know how the delay affects our system.
3. The user is not consistent with his feedback and it can vary over time, at first providing a substantial amount of feedback and after a while just giving occasional suggestions, so we would like to know if it is important to take into account such behavior in our system.
4. Under the above mentioned conditions, can we still guarantee convergence and if so, under which conditions?

This paper addresses most of these issues. The rest of the paper is organized as follows. Section 2 describes related work. In Section 3 we provide the theory required to answer the previous questions. Section 4 describes the experimental results with a mobile robot performing navigation tasks and Section 5 concludes and suggests future research directions.

## 2 Related Work

There is a large amount of literature describing reinforcement learning techniques in robotics. In this section we only review some of the most closely related work to our proposal. Reinforcement learning including feedback has been considered in some approaches [23,11,6,10,12]. Hardware devices such as joysticks, keyboards, among others are used to provide feedback. Other closely related approaches use voice commands to teach how to perform a task, as presented in [19,24]. In [19] a human instructor demonstrates how to do a task and gives instruction by speech. But, verbal instructions are very similar to control structures of a programming language that can be difficult to give by common people, and the learning is focused on learning by demonstration. By contrast, the method that we propose uses a more natural spoken interaction and uses reinforcement learning in addition of demonstration. In [24] an Actor-Critic algorithm based on IHDR (Hierarchical Discriminant Regression) is presented. A teacher sets the robot's arm in different positions, each one is named with a particular word (command) and the complete sequence of positions is named too. A position or a sequence of positions can then be invoked with its identifier. A feedback is given during the training phase, the teacher uses a bumper to mark the positions and to provide some reinforcements, so is necessary an interaction with hardware. In this paper, we propose a method without any special hardware requirement of the robot and with a more natural interaction approach. Unlike other approaches, our method uses language in a more natural way to guide the demonstration, and uses reinforcement learning with verbal feedback as dynamic reward shaping.

Reward shaping has been used to accelerate reinforcement learning [15,9]. More recently, researchers have try to learn a reward shaping function [1,7,4,3,13], with designs of structural shaping (decomposition of a complex task in simple tasks) as [5,14,7,18], and with shaping that increases the complexity of a task over time as [5,14]. However, most of these methods require domain knowledge to design an adequate reward shaping function, which are static over time, or try to learn the functions with experience, which can take very long training times. Some authors have provide traces from a user that can be used to guide the reinforcement learning process [1,8]. Our method also uses traces provided by a teacher but can also incorporated verbal feedback during the learning process.

Contrary to other approaches, our method uses initial traces and feedback (commands and qualifiers) provide by a teacher through voice that can be given at any time during the learning process. It deals with continuous spaces and can reach good policies with moderated noisy feedback. In the following sections we describe in detail the proposed method.

## 3 Dynamic Reward Shaping

In this section we introduce our formal framework. Reinforcement learning problems can be modelled as Markov Decision Processes (MDPs). Given an MDP, $M = (S, A, T, R)$, what we want to learn is a policy, i.e., the best action to perform on each state to receive the maximum expected accumulated reward. The idea of reward shaping is to give additional rewards to a learning agent to guide its learning process and converge faster. In effect, the learning algorithm is running on a transformed MDP, $M' = (S, A, T, R')$, where $R' = f(R, s, s')$. Normally $f$ has been used as an additive function, i.e., $R' = R + F$, but in general, it does not need to be the case. So in the new MDP when an agent takes in state $s$ action $a$ and moves to $s'$ it receives a reward defined as $R + F$.

A significant advancement in the formalization of reward shaping was the development of potential-based reward shaping [15]. Here the idea is that the reward for executing a transition between two states is mainly the difference in the value of a potential function $\phi$ applied to each state. It can obtain optimal policies and solve some problems reported in [18] with shaping functions. In particular, the difference of potentials avoids creating loops between a sequence of states when an agent gain positive rewards that deviates it from the goal. The function $F$ is represented under this framework as: $F(s, s') = \gamma\phi(s') - \phi(s)$.

Potentials are adequate if they are given when the agent gets closer towards the goal and are normally defined by the user. Recently there has been some work on how to learn a potential, by learning a value function over a simplified problem and use this value function as reward shaping or by using an estimate of the distance to the goal, similarly as the underestimate measures used in search techniques [13]. In simple mazes this can be a Manhattan distance between the agent and the goal, however, they can produce wrong estimates if obstacles are ignored, as getting closer in straight line can actually take the agent away from the goal.

### 3.1 On-Line Feedback by the User

In our case, we assume that the shaping rewards are given by the user through voice commands. The user can qualify the behavior of the robot, for instance, good or bad, which are then translated into rewards, or can command the robot to perform a particular action. Algorithm 1 describes the pseudo-code of SARSA learning with dynamic rewards and other algorithms could be adapted as well.

**Algorithm 1.** SARSA learning with dynamic rewards.

```
Initialize $Q(s,a)$ arbitrarily
for each episode do
  Initialize $s$
  repeat
    for each step in episode do
      if User selects action $a$ in $s$ then
        Take action $a$, observe $r$, $s'$
        Let $r \leftarrow h(r,a)$ {where $h(r,a)$ is a bounded function that depends on the selected action}
      else
        Select an $a$ from $s$ derived from policy $Q$ (e.g., $\epsilon$–greedy)
        Perform action $a$, observe $r$, $s'$
      end if
      if user provides feedback $f$ on state $s'$ then
        Let $r \leftarrow g(r,f)$ {where $g(r,f)$ is a bounded function that depends on the provided feedback}
      end if
      Select $a'$ from $s'$ derived from policy $Q$ (e.g., $\epsilon$–greedy)
      $Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma Q(s',a') - Q(s,a)]$
      $s \leftarrow s'$;
    end for
  until $s$ is a terminal state.
end for
```

When the user is providing feedback on-line, there can be several cases:

**Permanent feedback:** If we assume that the user is continuously providing adequate feedback to the robot, then we can use results from difference of potentials. It is expected that, at least for many domains, the user has a clear idea of what represents progress towards the goal. We only scale-up the value function and learn the same policy.

**Sporadic feedback:** In our case, the user may provide voice commands, and consequently affect the rewards, but not all the time and not on the same state-action pairs. In this case we have sporadic rewards. This occasional rewards, depending on their frequency, will fade-out with time and with distance. However, if we have a reward persistently given in a particular state, this will create a new sub-goal. In this way, the user can guide the robot towards intermediate goals. If we have relatively frequent rewards all over the space, we can still converge towards the optimal policy as long as the expected rewards on each state follow a potential-based reward function. In general, this is not too difficult to have as we expect the user to have a clear idea of when the robot is actually progressing towards the goal.

**Noisy feedback:** Now suppose the user sometimes gives a wrong reward, gives it out of time or the speech recognition system misinterprets the command. In any case the agent receives a wrong reward. These situations can deviate the agent towards a wrong policy. In this case, the agent can still recover provided the wrong rewards are not given all the time and are stopped while there are good chances of following exploratory movements. Also, the user can try to correct the errors with additional feedback. In this sense, the user can always impose his own policy. If the rewards are given in average towards the goal during the learning process, then we can still produce an optimal policy as before, although it may take longer training than a strategy without reward shaping.

In all cases, our reward function is defined as: $R' = R + F$ where $F$ is a reward provided 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.

## 4 Experiments

In this section, we illustrate our approach using an autonomous mobile robot that learns navigation tasks in simulated environments. We used Player/Stage to simulate an autonomous mobile robot Pioneer 2 for the experiments. The robot has a frontal laser and a sonar in the rear part. And, we used Sphinx 3 to the speech recognizer based on the corpus DIMEx100 [17].

In order to learn a new task, the user first tells the robot what actions to do in order to complete the task. The user can provide one or more initial traces that are used as guidance to the robot. With these initial traces the robot follows a reinforcement learning process to try to improve the performance over these traces as they are expected to have errors and be far from optimal. During learning the robot receives rewards by the the traditional reward function defined in reinforcement learning but can also received additional rewards from user's feedback. In the experiments our reward function is given by: $R = R_{RL} + R_U$, where $R_{RL}$ is the traditional reinforcement learning reward function and $R_U$ is the reward associated with the user's voice commands. We used Algorithm 1 with eligibility traces for learning.

In our algorithm, the states are represented with information from the robot's sensors and are incrementally created while the robot is traversing the environment. When the readings of the sensors are sufficiently different from previous stored states a new state is created. We use Pearson's coefficient as similarity function and a threshold value to decide if the current sensor readings is sufficiently different from the current stored states to create a new state. Each state is associated with a set of discrete actions but our framework is able to produce continuous actions by combining discrete actions considering their current Q-values (more details of the representation and the way to produce continuous actions are given in [22]).

The language chosen to interact with the robot was Spanish. The vocabulary (with around 250 words) was composed of words which we later expanded in a second corpus to deal with short phrases. The words of interest were qualifiers and commands of actions. Part of it is showed in Table 1.

**Table 1.** Examples of vocabulary with a rough translation into English

| WORDS | SHORT PHRASES |
|---|---|
| Avanzar (forward) | Hacia adelante (move forward) |
| Regresar (backward) | Hacia atrás (move backwards) |
| Izquierda (left) | Gira a la izquierda (turn to your left) |
| Derecha (right) | Ve a tu derecha (go to your right) |
| Fin, Final (end) | Para ahí (stop there) |
| Bien (good) | Sigue así (keep like this) |
| Mal (bad) | Por ahí no (not that way) |
| Excelente (excellent) | Muy bien (very good) |
| Terrible (terrible) | Así no (not like that) |
| Objetivo (goal) | Hasta ahí (until there) |

In the experiments, we associated rewards to certain words of the vocabulary: +100 for reaching the goal (*objetivo*), +50 for "excellent" (*excelente*), +10 for "good" (*bien*), −50 for "terrible" (*terrible*), and −10 for "bad" (*mal*). Similar rewards were used to the rest of the vocabulary and the phrases.

We performed several robot navigation experiments. Figure 1 shows the different tasks that were taught to the robot. We compared the performance of the following settings:

1. SARSA($\lambda$)
2. SARSA($\lambda$) with user's feedback considering errors (around 20% noisy) of our current speech recognition system

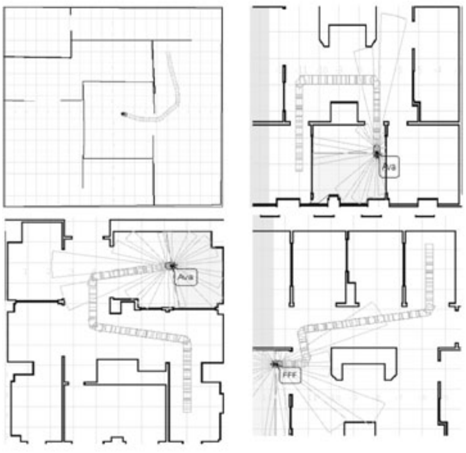

**Fig. 1.** Tasks 1-4 (left to right, up to down) that were taught to the robot

3. SARSA($\lambda$) with user's feedback and with initial guided traces considering errors (around 20% noisy) of our current speech recognition system
4. SARSA($\lambda$) with user's feedback and with initial guided traces without interpretation errors from the speech recognition system

Each experiment was repeated three times and averages are shown in the results. Figure 2 and Table 2 show the total times of the learning process for learning the four tasks. As can be seen, the number of episodes is roughly similar in the three cases (although it is smaller with feedback and initial traces) but the learning times are much faster when the algorithm uses feedback from the user (3.84 times faster than traditional reinforcement learning when we used feedback and 5.4 times faster when we used feedback and initial traces).

Table 3 shows to the left the number of interventions (feedback) provided by the user on each task. As can be seen in average the number of interventions is reduced in half if initial traces are given beforehand. Table 3 shows to the right a comparison between learning with perfect feedback (no mistakes from the speech understanding system) and with noisy (around 20% noisy) feedback for Task 2.

**Table 2.** Total number of episodes (left) and computing times (right) of the different experiments per task. RL = SARSA learning, RL + F = RL with feedback from the user, and RL + T + F = RL + F with the initial traces provided by the user.

| | RL | RL + F | RL + T + F |
|---|---|---|---|
| T1 | 13 | 9 | 6 |
| T2 | 6 | 7 | 7 |
| T3 | 12 | 12 | 7 |
| T4 | 7 | 12 | 11 |
| Avg | 9.5 | 10 | 7.75 |

| | RL | RL + F | RL + T + F |
|---|---|---|---|
| T1 | 103.93 | 19.59 | 12.85 |
| T2 | 66.4 | 15.1 | 13 |
| T3 | 100.65 | 38.2 | 18.09 |
| T4 | 99.1 | 23.43 | 24.61 |
| Avg | 92.54 | 24.08 | 17.13 |

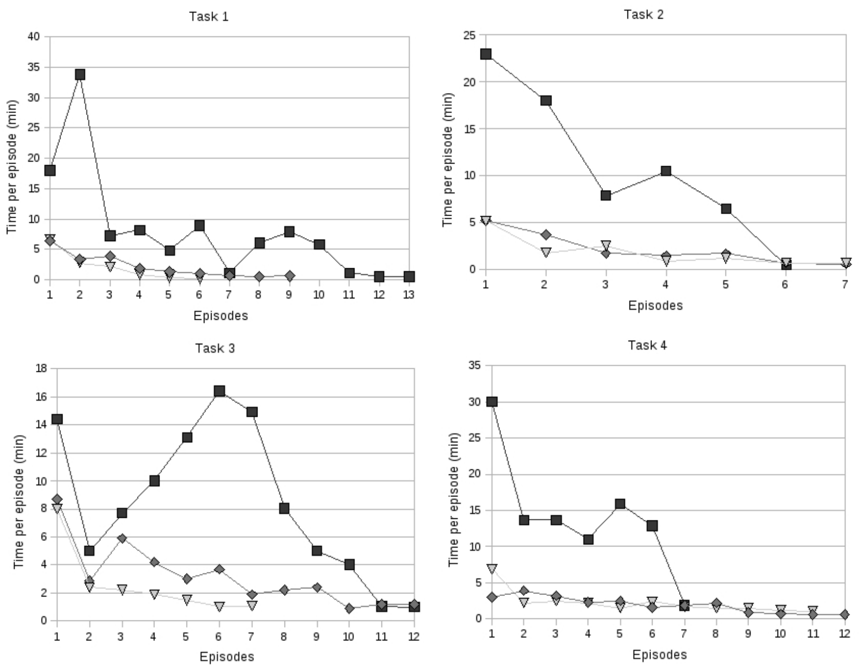

**Fig. 2.** Average times for learning the four tasks, using different variations of the algorithm: without traces and without feedback (square), with feedback (diamond), with traces and feedback (triangle). The $x$-axis has the number of episodes and the $y$-axis has the duration (minutes) of each episode.

**Table 3.** Number of interventions of the different experiments per task (RL + F = RL with feedback from the user and RL + T + F = RL + F with the initial traces provided by the user) on the left. Times per episode and number of interventions for Task 2 (on the right) both with the initial traces, first with noisy feedback, second with perfect feedback (PF).

| | RL + F | RL + T + F |
|---|---|---|
| T1 | 71 | 23 |
| T2 | 53 | 34 |
| T3 | 143 | 40 |
| T4 | 69 | 72 |
| Avg | 84 | 42 |

| | Times | Num. interventions |
|---|---|---|
| T2 | 13 | 34 |
| T2 PF | 7.6 | 39 |

As expected, the learning time is much faster with perfect feedback but also they both have an equivalent number of interventions. This shows that errors in the provided feedback can slow down the learning process but as shown in Table 2, our proposed approach is still between 3 to 6 times faster than normal reinforcement learning and is able to recover from noisy feedback.

As can be seen, by including voice feedback from the user significantly reduces the times of the learning process even with errors in the speech recognition system or in the commands provided. The best results are obtained when the user provides initial traces which also reduces the number of user's interventions.

## 5 Conclusions and Future Work

In this paper we described a new approach to incorporate feedback from the user into the learning process. The feedback is given by voice commands by a user which are translated into rewards and added to the reward function. Contrary to other approaches the feedback is given at any time during the learning process, can have errors and still converge faster than a traditional reinforcement learning approach. Our approach does not require any special hardware and we believe that it offers a more natural way to train robots new tasks in reasonable training times. We believe that by combining reinforcement learning with on-line feedback from the user is a promising line of research that offers sufficiently fast learning times and a natural instruction mechanism to be used in modern service robots.

There are several areas of improvement and future research work. In particular, we are starting to perform experiments with a manipulator. We would also like to include a command to undo some actions in order to converge faster. Also, we would like to extend our speech recognition system to provide more natural interactions and to consider different intentions, i.e., it is not the same to shout *stop*! to a robot heading towards a staircase than telling gently the robot to *stop* before receiving a new command.

## References

1. Abbeel, P., Ng, A.Y.: Apprenticeship Learning via Inverse Reinforcement Learning. In: 21st International Conference on Machine Learning (2004)
2. Conn, K., Peters, R.A.: Reinforcement Learning with a Supervisor for a Mobile Robot in a Real-World Environment. In: IEEE Computational Intelligence in Robotics and Automation (2007)
3. Dorigo, M., Colombetti, M.: Robot Shaping: Developing Autonomous Agents through Learning. Artificial Intelligence Journal 2, 321–370 (1993)
4. Grzes, M., Kudenko, D.: Learning Shaping Rewards in Model-based Reinforcement Learning. In: Workshop on Adaptive Learning Agents ALA-AAMAS (2009)
5. Gullapalli, V.: Reinforcement Learning and its Application to Control. Ph.D. Thesis. University of Masachussetts (1992)
6. Iida, F., Tabata, M., Hara, F.: Generating Personality Character in a Face Robot through Interaction with Human. In: 7th IEEE International Workshop on Robot and Human Communication, pp. 481–486 (1998)
7. Konidaris, G., Barto, A.: Autonomous Shaping: Knowledge Transfer in Reinforcement Learning. In: 23rd International Conference on Machine Learning (2006)
8. Knox, W.B., Stone, P.: Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. In: 9th International Conference Autonomous Agents and Multiagent Systems (2010)

9. Laud, A.: Theory and Application of Reward Shaping in Reinforcement Learning. PhD. Thesis. University of Illinois (2004)
10. Lockerd, T.A., Breazeal, C.: Reinforcement Learning with Human Teachers: Evidence of Feedback and Guidance with Implications for Learning Performance. In: 21st National Conference on Artificial Intelligence (2006)
11. Lockerd, T.A., Hoffman, G., Breazeal, C.: Real-Time Interactive Reinforcement Learning for Robots. In: Workshop on Human Comprehensible Machine Learning (2005)
12. Lockerd, T.A., Hoffman, G., Breazeal, C.: Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots. In: 15th IEEE International Symposium on Robot and Human Interactive Communication, pp. 352–357 (2006)
13. Marthi, B.: Automatic Shaping and Decomposition of Reward Functions. In: 24th International Conference on Machine Learning, pp. 601–608 (2007)
14. Mataric, M.: Reward Functions for Accelerated Learning. In: 11th International Conference on Machine Learning, pp. 182–189 (1994)
15. Ng, A.Y., Harada, D., Rusell, S.: Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping. In: 16th International Conference on Machine Learning, pp. 278–287 (1999)
16. Peters, J., Vijayakumar, S., Schaal, S.: Reinforcement Learning for Humanoid Robotics. In: 3rd IEEE-RAS International Conference on Humanoid Robots (2003)
17. Pineda, L.: Corpus DIMEx100 (Level T22). DIME Project. Computer Sciences Department. IIMAS, UNAM, ISBN:970-32-3395-3
18. Randlov, J., Alstrom, P.: Learning to Drive a Bicycle using Reinforcement Learning and Shaping. In: 15th International Conference on Machine Learning, pp. 463–471 (1998)
19. Rybski Paul, E., Kevin, Y., Jeremy, S., Veloso Manuela, M.: Interactive Robot Task Training through Dialog and Demonstration. In: ACM/IEEE International Conference on Human Robot Interaction, pp. 255–262 (2007)
20. Smart, W., Kaelbling, L.: Effective Reinforcement Learning for Mobile Robots. In: IEEE International Conference on Robotics and Automation (2002)
21. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1999)
22. Tenorio-Gonzalez, A.C.: Instruction of Tasks to a Robot using on-line Feedback provided by Voice. M.S. Thesis (2010)
23. Wang, Y., Huber, M., Papudesi, V.N., Cook, D.J.: User-Guided Reinforcement Learning of Robot Assistive Tasks for an Intelligent Environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2003)
24. Zhang, Y., Weng, J.: Action Chaining by a Developmental Robot with a Value System. In: 2nd International Conference on Development and Learning (2002)