In: IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1341-1348, Killarney, Ireland (2015)

Interactive Reinforcement Learning through Speech Guidance in a Domestic Scenario

Francisco Cruz, Johannes Twiefel, Sven Magg, Cornelius Weber, and Stefan Wermter
Knowledge Technology Group, Department of Informatics, University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
Email: {cruz, twiefel, magg, weber, wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

***Abstract*—Recently robots are being used more frequently as assistants in domestic scenarios. In this context we train an apprentice robot to perform a cleaning task using interactive reinforcement learning since it has been shown to be an efficient learning approach benefiting from human expertise for performing domestic tasks. The robotic agent obtains interactive feedback via a speech recognition system which is tested to work with five different microphones concerning their polar patterns and distance to the teacher to recognize sentences in different instruction classes. Moreover, the reinforcement learning approach uses situated affordances to allow the robot to complete the cleaning task in every episode anticipating when chosen actions are possible to be performed. Situated affordances and interaction allow to improve the convergence speed of reinforcement learning, and the results also show that the system is robust against wrong instructions that result from errors of the speech recognition system.**

I. Introduction

The field of robotics has shown considerable progress in the last years allowing robots to be present in diverse scenarios, from industrial environments where they are nowadays established to domestic environments where their presence is still limited [1]. In this paper we propose a domestic scenario where a robot has to perform a task which consists of cleaning a table assisted by a minimal degree of external guidance. A robot working in a domestic scenario can benefit from human expertise on how to perform a particular task [8] [9] [10]. Therefore, in our scenario the robot receives spoken advice from a human trainer which is recognized by an automatic speech recognition (ASR) system. This way of giving instructions is natural for humans, but we need to control the probability of supplying feedback by the teacher, as humans can decide if they provide an instruction in a given situation. Hence, we use an artificial agent with full knowledge about the task to provide the spoken advice.

In neural networks there are mainly three different learning paradigms, which are supervised, unsupervised and reinforcement learning [2]. In our scenario the robot has no previous knowledge on how to perform the task and it needs to explore its environment. Therefore the apprenticeship process is carried out with reinforcement learning (RL).

RL is a learning approach supported by behavioral psychology where an agent interacts with its environment trying to find an optimal policy to perform a particular task. In every time step, the agent performs an action reaching a new state and obtains either a reward or a punishment. The agent tries to maximize the obtained reward by choosing the best action in a given state [3].

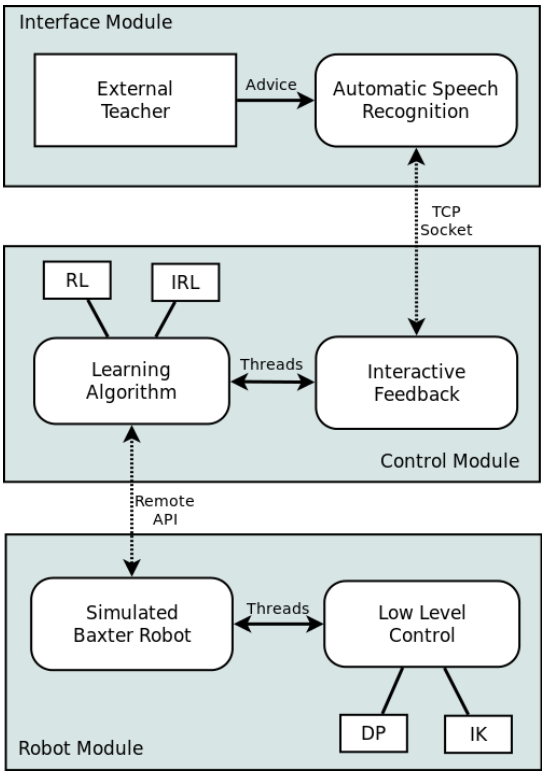


Fig. 1. System architecture in three levels. At the top is the interface module which interacts with the external teacher, in the middle is the control module and at the bottom the robot module where the actions are performed.

One RL problem that still remains open is the time spent by an RL agent during learning. It often requires excessive time to find a proper policy [4], mainly due to the large and complex state action space which leads to excessive computational costs. In this work, besides external guidance, we use situated affordances which are a generalization of the affordance concept [6] which lately has been used successfully in robotics [7]. Situated affordances are implemented by a deep multilayer perceptron allowing us to estimate either the robot's next state or if the affordance is temporally unavailable.

Therefore, the following work presents an integrated system

(c) 2015 IEEE. Personal use of this material is permitted.

to teach a robot to perform a domestic cleaning task using an external trainer with occasional spoken instructional feedback. The system architecture consists of three modules which are shown in Figure 1. At the top there is the interface module where the external trainer provides a voice stream which is processed by the ASR system and sent to the control module in the middle. The control module runs the learning algorithm that is able to perform autonomous RL and IRL generating choices for actions. These are passed to the robot module to be executed by a simulated Baxter robot in the V-REP simulator [11] combining two different approaches for low level control, namely direct planning and inverse kinematics.

II. Reinforcement Learning and Interactive Guidance

A prominent strategy to improve the speed of convergence in RL is to use external trainers to provide guidance in specific states during the learning process. Early research on this topic can be found in [12] where the author shows that external guidance plays an important role in learning tasks performed by both humans and robots by decreasing the time needed for learning. Furthermore, in large spaces where a complete search through the whole search space is unfeasible, the trainer may lead the apprentice to explore more prominent areas at early stages. Additionally, trainers may also help the learner to avoid getting stuck in local maxima.

So far, diverse strategies have been presented to provide guidance in RL, such as learning by imitation [13], demonstration [14] [15] [16], and feedback [17] [18] [19]. In particular in learning by feedback two main approaches are distinguished: reward and policy shaping. Whereas in reward shaping an external trainer is able to evaluate how good or bad the performed actions by the RL agent are [18] [20], in policy shaping the action proposed by the RL agent can be replaced by a more suitable action chosen by the external trainer before it is executed [17]. In both cases, an external trainer gives interactive feedback to the apprentice agent to encourage it to perform certain actions in certain states to reach either a better policy or a faster performance.

In this work, we use policy shaping through action selection guidance which means that the action to perform may be given by an external trainer that has prior knowledge about the task (see figure 2). Diverse information sources can be employed to obtain feedback from, for instance, a person, another robot, or an simulated agent.

In our human-robot scenario it is desired to keep the rate of interaction with an external trainer as low as possible; otherwise reinforcement learning converts into supervised learning. Also, the consistency or quality of the feedback should be considered to determine whether learning is still improving given that the external teacher could also make mistakes [21].

We presented in [22] a method which allows to improve the speed of convergence of an RL agent using affordances and interaction. Affordances limit the number of possible actions in some states and can reduce the computational complexity of RL. The interactive feedback was provided by an artificial agent acting as an external trainer instead of a human. This agent had previously learned the task and it had to provide feedback with a probability of 30%. The results show a reduction in the number of required episodes during the training as well as a reduction in the number of actions performed in each episode. However, in our previous work, affordances were given as prior knowledge and were not learned automatically.

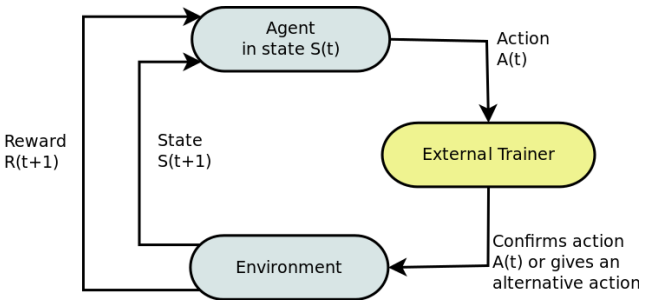


Fig. 2. Our approach to interaction between a robotic agent and an external trainer by feedback. In this case, the external trainer is able to change a selected action to be performed in the environment.

III. Contextual Affordances

Affordances are often seen as opportunities for action of an agent (a person, an animal, a robot, or an organism). The original concept comes from ecological psychology and was proposed by Gibson [6]. For instance, a glass and a bed afford different actions to a person who is able to grasp the glass and lie down on the bed, but cannot do it the other way around. Thus, an agent is able to determine some object affordances beforehand and the caused effect after a specific action is performed with an object. In Gibson's book many diverse examples are given but no concrete, formal definition is provided.

Even nowadays, we find marked differences among ecological psychologists about the formal definition of affordances [23] and these discrepancies could even be stronger between them and AI scientists [24] [25]. In the following subsections we propose a formal computational definition based on the original concept of Gibson and then an extension considering an additional context variable.

A. Affordances

Affordances have been particularly useful to establish relationships between actions performed by an agent with available objects. We use them in a way to represent object/action information. They represent neither agent nor object characteristics, but rather the characteristics of the relationship between them. In [26] an affordance is defined as the relationship between an object, an action, and an effect as the triplet:

$$Affordance = (Object, Action, Effect) \quad (1)$$

Figure 3 shows the relationship between the previous components, where objects are entities which the agent is able to

(c) 2015 IEEE. Personal use of this material is permitted.

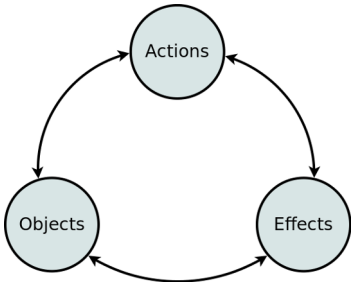


Fig. 3. Affordances as relations between objects, actions, and effects. Objects are entities which the agent is able to interact with, actions represent the behavior that can be performed with the objects, and the effects are the results caused by applying an action [26].

Fig. 4. The book affords grasping as long as the agent's current state allows this action to be performed. In the shown scenario it is not feasible to use that affordance since both hands are already occupied, but once one hand is free the affordance can be used again.

interact with, actions represent the behavior or motor skills that can be performed with the objects, and the effects are the results of an action involving an object [24] [27].

It is also important to note that the object in equation 1 can also be a place or a location, for instance, a hill affords climbing. From here onwards, we employ the term *object* to refer to the affordance component but we consider also locations.

B. Situated Affordances

If an affordance exists and the agent has knowledge and awareness of it, the actual, next step is to determine if it is possible to utilize it considering the agent's current state. For instance, let us consider the following scenario: a cup affords grasping, as does a book, but in case we have an agent with both hands occupied (e.g. with one cup in each hand) then the agent cannot grasp the book anymore or in other words, the affordance is temporarily unavailable. This situation is depicted in figure 4. This does not mean that the affordance does not exist, but to the contrary, the affordance is still present but cannot be used by the agent in that particular situation due to its current state.

Kammer et al. [32] proposed to consider the dynamics in the environment in which the object was embedded rather than

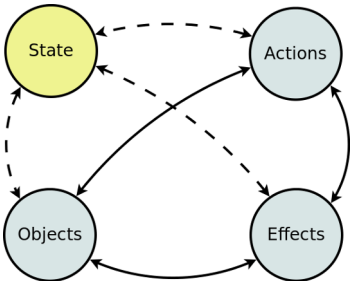


Fig. 5. Situated affordances as relations between state, objects, actions, and effects. In this case the state is the agent's current condition. According to this state a different effect could be produced in different occasions.

the agent's dynamic state. The awareness of this extra variable is called situated affordances. Even though a formal definition was provided, neither applications nor results are shown in this work. Nevertheless, we use the same concept to address the problem when the agent's state is dynamic. Thus, we propose a model where the current state of an agent is also considered for the effects of an action performed with an object. The following quartet shows this model:

$$\textit{Situated Affordance} = \textit{(State, Object, Action, Effect)} \quad (2)$$

For instance, given two affordances utilizing the same action a and the same object o, but from a different agent's state $s_1 \neq s_2$. When action a is performed different effects $e_1 \neq e_2$ are generated and it is unfeasible to establish differences between these affordances, i.e. $e_1 = (a, o)$ and $e_2 = (a, o)$. Hence, to deal with the current states $s_1 \neq s_2$, an agent distinguishes each case and learns them at the same time since each situated affordance would be defined by $e_1 = (s_1, a, o)$ and $e_2 = (s_2, a, o)$ establishing clear differences between them with enough information [32].

Figure 5 shows the relationship between object, action, effect, and the agent's current state. This model allows us to determine beforehand when it is possible to apply an affordance using an artificial neural network (ANN) to learn the relationship with the state, the action, and the object as inputs and the effect as output. In chapter VI we will describe in detail the neural network architecture used to anticipate the effect.

IV. Cleaning Scenario

A detailed domestic scenario has been defined [22] on a table cleaning task. To achieve this task we have defined objects, locations, and actions. The task includes a robot standing in front of a table which should be cleaned with a sponge. The table is divided into two zones to be cleaned. In one of these zones a cup is placed which has to be moved during the task execution to complete it successfully. A third location called *home* is also considered to refer to the initial arm position as well as the place where the sponge is kept. We have defined four actions: *get* an object, *drop* an object, *go* to a location, and *clean* which cleans the table portion where

(c) 2015 IEEE. Personal use of this material is permitted.

TABLE I
LIST OF DEFINED OBJECTS, LOCATIONS AND ACTIONS FOR CLEANING-TABLE SCENARIO.

Objects	Locations	Actions
sponge	left	get <object>
cup	right	drop <object>
	home	go <location>
		clean

the robot arm is currently placed. Table I shows a summary of available objects, locations, and actions.

The defined cleaning scenario has 46 states; each one is obtained considering the following four variables:

i. the robot's hand position,
ii. the object held in the hand, or free,
iii. the position of the cup,
iv. the current condition of the two locations.

Given the defined actions, objects and locations, we are able to set the presence of four different situated affordances which allow us to determine if objects are *graspable, droppable, movable,* or *cleanable* according to the robot's current state.

To make the scenario more complex, we created variations for every combination of action and object/location, expanding it to 33 domain-specific instructions belonging to 8 different classes. For instance, the instruction `get the sponge` could also be stated as `pick up the sponge`, `take the sponge`, `grasp the sponge`, or `lift the sponge`, but all of them belong to the same class.

The table cleaning task is carried out by a Baxter robot in a simulated environment using the V-REP simulator [11]. All actions are performed using only one arm which has seven degrees of freedom (DoF). Figure 6 shows the scenario while the Baxter robot is cleaning the table using the sponge. The Baxter robot has as end effector a vacuum cup, also called suction pad. We did not employ a gripper to grasp the object since the main focus in this work was to learn the right sequence quickly. Moreover, to reach the defined locations direct planning was used and then afterwards inverse kinematics for low-level control was used to grab objects.

V. AUTOMATIC SPEECH RECOGNITION

The given scenario originates from the Human-Robot Interaction (HRI) domain. For this reason, we do not only employ a humanoid robot as the learner, but there is also a humanoid teacher. Since the human way of instructing a robot is employing speech, the teacher also uses speech to instruct the learning robot by providing pre-recorded audio data that was spoken by a human. To understand the verbal commands, the apprentice processes audio data and recognizes the given guidance by applying an ASR system.

The ASR system we employ for our approach is based on *Google Voice Search* [28] which is a cloud-based ASR service processing audio data captured by a local microphone and generating hypotheses for the corresponding text representation. *Google Voice Search* utilizes well-trained acoustic

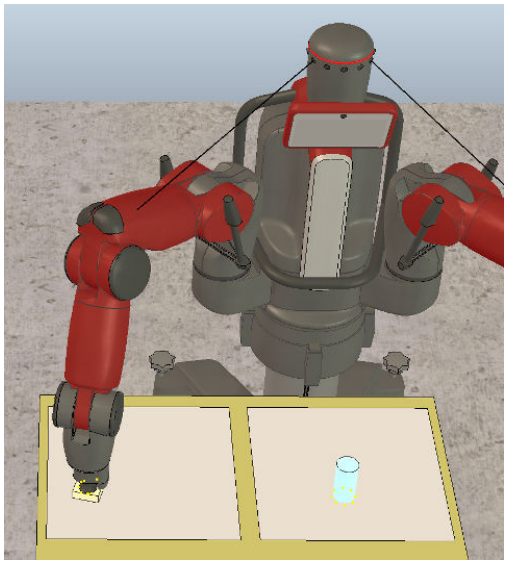


Fig. 6. A simulated Baxter robot performs the actions in the environment which is created in the V-REP simulator. The cleaning scenario consists of three locations, two objects, and four actions.

models based on large amounts of audio data collected [29]. As *Google Voice Search* is generally applied in web searches, the involved language models are optimized for this task. The given HRI scenario differs from this field as robot instructions are verbalized which are not the first preference in web search-based ASR hypotheses. A possibility to overcome this issue is to integrate a local open-source ASR system which can be configured by providing a domain-specific language model for the given HRI scenario. However, the acoustic models employed by local open-source ASR systems provide a lower quality due to the lower amount of training data available during training.

To overcome the issues of either weak acoustic models or out-of-domain language models we developed a post-processing technique to fit the ASR hypotheses provided by *Google Voice Search* to the given HRI domain. To be able to exploit the quality of the well-trained acoustic models employed by *Google Voice Search*, the ASR hypothesis is converted to a phonemic representation employing the *SequiturG2P* grapheme-to-phoneme (G2P) converter [30] trained on the *CMUdict 0.7a*[1] dictionary which contains words and their corresponding phoneme sequence representation. The G2P converter is capable of creating a phoneme sequence for unknown words based on the provided training data and so overcomes the issue of unknown words contained in the ASR hypothesis provided by *Google Voice Search*.

For the given HRI scenario, a fixed set of robot commands is defined and represented by a list of sentences. To receive the best-matching hypothesis out of the list of sentences, the phonemic representation of the ASR hypothesis is compared to the phonemic representations of each sentence in the list. For this task, the Levenshtein distance [31] is employed to

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

(c) 2015 IEEE. Personal use of this material is permitted.

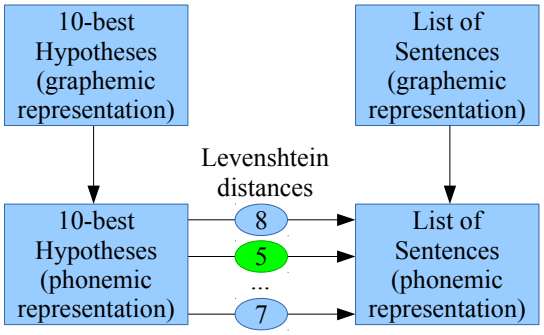


Fig. 7. Functional principle of the ASR system. The left side shows the ASR hypotheses provided by Google and the right side contains the list of sentences for the given HRI scenario. In the middle, the Levenshtein distances are calculated.

calculate the difference between phoneme sequences. After calculating the Levenshtein distance between the ASR hypothesis and each sentence of the list, the sentence possessing the shortest distance is chosen as the best matching result. To improve the technique, the Levenshtein distance is calculated for the ten best hypotheses provided by *Google Voice Search*. Figure 7 summarizes the mentioned functional principle.

VI. Proposed Model

A general overview of the system architecture is shown in figure 1. This section describes the proposed model considering aspects such as our IRL approach and how situated affordances are implemented with an ANN architecture to estimate the robot's next state.

A. Interactive Reinforcement Learning Approach

Since RL is used, most of the time the robot performs actions autonomously by exploring the environment, but when guidance is delivered this is sent to the robot via speech recognition. Hence the robot takes advantage of this advice in selected time steps during a learning episode and performs the suggested actions to complete the task in shorter time by performing fewer actions.

In the learning module we allow the robot to perform actions considering transitions from state-action pair to state-action pair rather than transitions from state to state only. Therefore, we implement the on-policy method SARSA [33] to update every state-action value according to equation 3:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (3)$$

where s_t and s_{t+1} are the current and next state respectively, a_t and a_{t+1} are the current and next action, Q is the value of the action-state pair, r_{t+1} the obtained reward, α is the learning rate and γ the discount factor. The reward function delivers a positive reward of 1 to the agent every time it reaches the final state, otherwise it delivers a reward of 0. Equation 4 shows the reward function defined:

$$r(s) = \begin{cases} 1 & \text{if s is the final state} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Furthermore, we use the ϵ-greedy method for action selection with the following parameters $\alpha = 0.3$, $\gamma = 0.9$, and $\epsilon = 0.2$. Therefore, most of the time the next action is determined as shown in equation 5:

$$a_{t+1} = \underset{a \in a_s}{\operatorname{argmax}}\, Q(s_t, a) \quad (5)$$

where s_t is the current state at time t, and a_s corresponds to a subset of all available actions. The subset of actions is determined based on the situated affordances (see equation 2 and/or figure 5). This way, it is possible to anticipate whether the action is performable with an object in a particular state which is called effect. This neural architecture will be explained in the next subsection.

Our external trainer consisted of an artificial agent that had full knowledge about the task. Therefore it was able to deliver selected interactive feedback to the robot using ASR at certain times during the learning process. We used the *advise method* defined in [21] with probability of feedback $\mathcal{L} = 0.2$ and consistency of feedback $\mathcal{C} = 0.9$ as interaction parameters.

B. Situated Affordances with Deep Neural Architecture

It has been shown that a multilayer feedforward neural network (MLP) with only one hidden layer and a sufficient number of neurons in this layer is able to approximate any continuous non-linear function with arbitrary precision [34] [35] [36]. Nevertheless, MLPs with only one hidden layer may need an exponential number of neurons in order to reach a particular degree of precision [37]. Besides that, in the last years deep neural architectures have become a topic of interest within the research community due to their distributed and sparse representation which allows to tackle problems in a similar fashion as the human brain does [38].

Therefore, to learn the relationship between inputs and outputs in situated affordances we implement a deep multilayer perceptron (DMLP) which is a feedforward network with more than one hidden layer as proposed in [39]. As inputs we use the agent's current state, the action, and the object giving a code number to every state and every action plus object, the two latter together; as output we use the next state or -1 to indicate that it is not feasible to perform the action with the object in the current state.

Training data were obtained considering all possible states mentioned in section IV as well as the instructive classes taking into account the combination of actions and objects/locations. This led to 368 data for the training process.

The final architecture consists of 46 neurons in the first hidden layer and 8 neurons in the second one. Both hidden layers have sigmoid transfer functions and the output layer has one neuron with a linear transfer function, as shown in figure 8. The number of neurons selected in every hidden layer is empirically determined related to our scenario.

(c) 2015 IEEE. Personal use of this material is permitted.

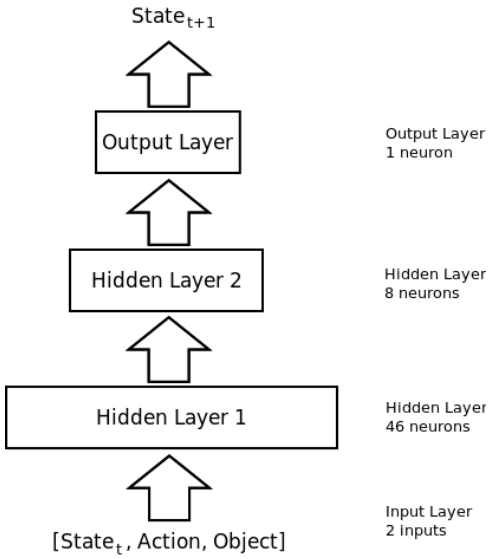
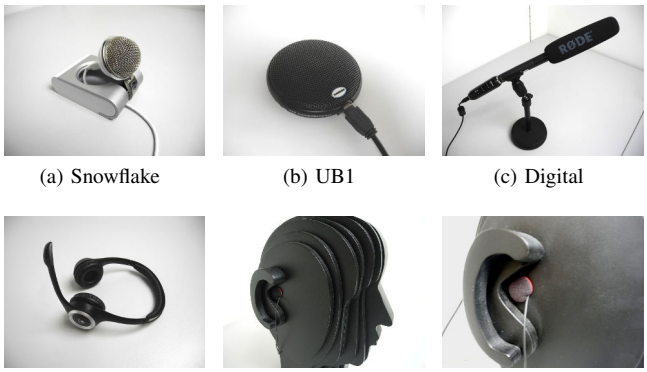


Fig. 8. Deep multilayer perceptron used to determine situated affordances. Hidden layers used sigmoid transfer functions and the output layer used a linear transfer function.



(a) Snowflake (b) UB1 (c) Digital

(d) Headset (e) Pro1 (f) Pro1(Details)

Fig. 9. Microphones used in the experiments

A general problem during the training process of a deep neural network is the vanishing gradient. For first order gradient-based methods a second problem of getting stuck can arise due to the error surface possessing large plateaux [40]. To overcome these issues, we use Nguyen-Widrow weight initialization [41] and the second order training method Levenberg-Marquardt due to better performance shown in [42].

Algorithm 1 shows the IRL approach using situated affordances, interaction and speech recognition. The conditional statement starting in line 19 represents the fact that the external teacher delivers advice and changes the next action a_{t+1} by formulating a verbal instruction that is processed by the ASR system. Conditions in lines 8 and 18 represent the response of the neural network about the feasibility of performing the action in the current state which is called situated affordance (SA in the algorithm).

Algorithm 1 Interactive reinforcement learning approach using situated affordances, interaction and speech recognition

Require: Previous definition of states and actions

1: Initialize $Q(s,a)$ arbitrarily
2: **repeat**
3: **if** $rand(0,1) <= \epsilon$ **then**
4: Choose a_t randomly from A
5: **else**
6: Choose a_t according to $a = \underset{a=a_s}{\operatorname{argmax}}\ Q(s,a)$
7: **end if**
8: **until** $SA(a_t, o, s_t) <> -1$
9: **repeat**
10: Take action a_t
11: Observe reward r_{t+1} and next state s_{t+1}
12: **repeat**
13: **if** $rand(0,1) <= \epsilon$ **then**
14: Choose a_{t+1} randomly from A
15: **else**
16: Choose a_{t+1} according to $a = \underset{a=a_s}{\operatorname{argmax}}\ Q(s,a)$
17: **end if**
18: **until** $SA(a_{t+1}, o, s_{t+1}) <> -1$
19: **if** $rand(0,1) <= feedbackProbability$ **and** $rand(0,1) <= consistencyProbability$ **then**
20: get *advice* from teacher voice using ASR
21: **if** $SA(a_{t+1}, o, s_{t+1}) <> -1$ **then**
22: $a_{t+1} \leftarrow advice$
23: **end if**
24: **end if**
25: $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$
26: $s_t \leftarrow s_{t+1}$
27: $a_t \leftarrow a_{t+1}$
28: **until** s is terminal

VII. Experimental Results and Discussion

To carry out the table cleaning task we consider different microphones to measure how the hardware affects quality in the ASR system and consequentially in the IRL approach. Therefore, we made simultaneous recordings using 5 different kinds of microphones and evaluated the answers of our ASR system. Afterwards, we made the scenario more difficult by positioning the microphones in a distance of 1m away from the speaker, which leads to the necessity of increasing the strength of the audio signal to compensate for the lower volume of the speech instructions and with this also increasing the level of environmental noise contained in the audio signal. As a hypothesis, we claim that more noisy audio data leads to worse ASR performance and so we can measure the robustness of the learning system by providing incorrect instructions. The microphones were *Snowflake, UB1, Digital, Headset* and *Pro1* which are shown in figure 9. *Snowflake*'s polar pattern is cardioid, *UB1* is omnidirectional, *Digital* is supercardioid, *Headset* is unidirectional, and the *Pro1* is omnidirectional. Only 16kHz, mono channel audio data was utilized.

(c) 2015 IEEE. Personal use of this material is permitted.

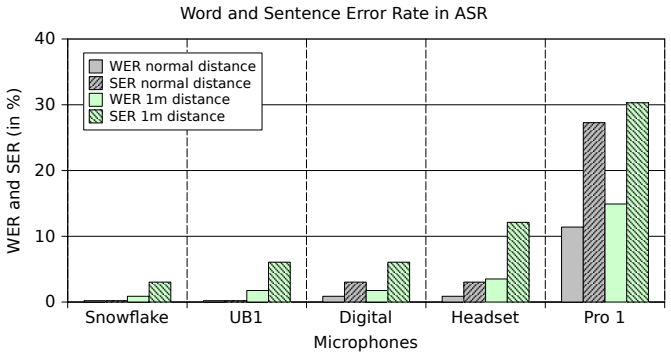


Fig. 10. Response of the ASR system to the list of sentences using different microphones at normal and at 1m distance. WER and SER are shown as percentages.

TABLE II
WORD AND SENTENCE ERROR RATE (%) IN ASR FOR ALL MICROPHONES USED AT NORMAL AND AT 1M DISTANCE.

Microphone	Normal distance		1m distance	
	WER	SER	WER 1m	SER 1m
Snowflake	0	0	0.877	3.03
UB1	0	0	1.754	6.061
Digital	0.877	3.03	1.754	6.061
Headset	0.877	3.03	3.509	12.121
Pro1	11.404	27.273	14.912	30.303

The response of the ASR module for the domain-specific language model measured in Word Error Rate (WER) and Sentence Error Rate (SER) is shown in figure 10 and in table II as percentages for normal distance and 1m distance. In this context, normal distance means that the microphone is placed in its normal working position depending on its characteristics. The SER depends on the sentence accuracy S_{Acc} as shown in the following equation:

$$SER = 1 - S_{Acc} \tag{6}$$

We observed that the microphone with the best results working with and without noise is *Snowflake* and the microphone with the worst result in both cases is *Pro1*.

To test the learning module three different set-ups were implemented: first the robot working autonomously, second the robot working with advice taken from the *Pro1* microphone, and third the robot working with advice taken from the *Snowflake* microphone. The two latest set-ups were run with the best and the worst microphone performances in the domain-specific language model with the aim of testing how influential the quality of the microphone was in improving the speed of convergence in IRL. In both cases, microphones at a distance of 1m away from the teacher were used.

Each set-up was carried out 100 times and the results were averaged. Figure 11 shows the average number of actions performed during 100 episodes. The y axis is truncated at 200 actions to highlight the difference between the set-ups. In each episode, the cleaning task was always finished because of situated affordances which allowed to avoid performing

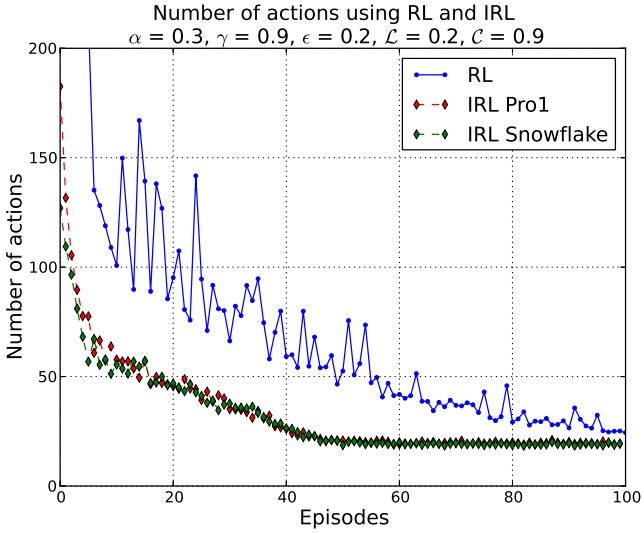


Fig. 11. Average number of actions performed to finish the task using an RL agent (blue) and an IRL agent with two different microphones (red and green). Despite of the differences in the hardware quality, the IRL approaches show improvement compared to the RL approach.

actions that, according to the robot's current state, were not feasible.

In figure 11, we show that both IRL approaches perform better than RL working autonomously, but there is no significant difference between IRL with *Pro1* and *Snowflake* microphones. To analyse this, we defined the actual interactive feedback rate ($\mathcal{I}$) which is the percentage of steps where a correct instruction was properly received by the agent:

$$\mathcal{I} = \mathcal{L} * \mathcal{C} * S_{Acc} \tag{7}$$

We computed $\mathcal{I}$ from the feedback probability ($\mathcal{L} = 0.2$), consistency probability ($\mathcal{C} = 0.9$), and the S_{Acc}. Given $SER \in [3.03\%, 30.303\%]$ the results were $\mathcal{I} = 12.55\%$ for *Pro1* and $\mathcal{I} = 17.45\%$ for *Snowflake*. These small values are already large enough for the agent to benefit from interaction. This is consistent with a recent study where it is shown that large improvements of RL by IRL are already achieved at low interaction rates [43]. In fact, it is possible to observe in figure 11 that there is a small variation in the first ten episodes but in the following episodes variations get even smaller. This leads to a system which is able to perform the task properly and which is robust for a variety of audio hardware.

VIII. CONCLUSIONS AND FUTURE WORK

We have shown ASR to be an effective method to work in IRL scenarios to improve the speed of convergence of RL agents. For scenarios where a human would verbally instruct a robot during IRL, our results indicate that interaction helps to increase the learning speed robustly even with an impoverished ASR system. Moreover, situated affordances allowed the agent to complete the RL task in every episode efficiently.

As future work, we will consider an architecture with free speech recognition which would allow us to move our

(c) 2015 IEEE. Personal use of this material is permitted.

domestic-cleaning scenario to work with human external trainers as well as a humanoid robot to perform the actions.

ACKNOWLEDGMENT

The first author gratefully acknowledges the support by *Universidad Central de Chile* and *CONICYT*.

REFERENCES

[1] T. S. Tadele, T. de Vries, and S. Stramigioli, *The safety of domestic robotics: A survey of various Safety-Related publications*, in IEEE Robotics & Automation Magazine, Vol. 21, no. 3, pp. 134–142, 2014.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 2nd printing edition, Springer, 2011.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, A Bradford Book, 1998.

[4] W. B. Knox and P. Stone, *Interactively shaping agents via human reinforcement: The TAMER framework*, in Proceedings of The Fifth International Conference on Knowledge Capture, pp. 9–16, 2009.

[5] J. Kober and J. Peters, *Reinforcement Learning in Robotics: A Survey*, in Reinforcement Learning, Vol. 12, Springer Berlin Heidelberg, pp. 579–610, 2012.

[6] J. J. Gibson, *The Ecological Approach to the Visual Perception*, Boston Houghton Mifflin, 1986.

[7] T.E. Horton, A. Chakraborty, and R. St. Amant, *Affordances for robots: a brief survey*, in AVANT: Journal of Philosophical-Interdisciplinary Vanguard, Vol (2), pp. 70–84. 2012

[8] A. L. Thomaz and C. Breazeal, *Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent*, in Proceedings of RO-MAN 2007 IEEE, pp. 720–725, 2007.

[9] H. B. Suay and S. Chernova, *Effect of human guidance and state space size on interactive reinforcement learning*, in Proceedings of RO-MAN 2011 IEEE, pp. 1–6, 2011.

[10] W. Knox, B. Glass, B. Love, W. Maddox, and P. Stone, *How Humans Teach Agents*, in Proceedings of International Journal of Social Robotics, Vol. 4, Issue 4, pp. 409–421, 2012.

[11] E. Rohmer, S. P. N. Singh, and M. Freese, *V-REP: A versatile and scalable robot simulation framework*, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2013), pp. 1321–1326, 2013.

[12] L. J. Lin, *Programming Robots Using Reinforcement Learning and Teaching*, In AAAI. pp. 781–786, 1991.

[13] J. P. Bandera, J. A. Rodríguez, L. Molina-Tanco, and A. Bandera, *A Survey of Vision-Based Architectures for Robot Learning by Imitation*, in International Journal of Humanoid Robotics, Vol. 09, No. 1250006, pp. 1–40, 2012.

[14] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, *Robot Learning From Demonstration by Constructing Skill Trees*, in The International Journal of Robotics Research, Vol. 31, No. 3, pp. 360–375, 2012

[15] L. Rozo, P. Jiménez, and C. Torras, *A Robot Learning From Demonstration Framework to Perform Force-Based Manipulation Tasks*, in Intelligent Service Robotics, Vol. 6, No. 1, pp. 33–51, 2013.

[16] J. Peters, J. Kober, K. Mülling, O. Krämer, and G. Neumann, *Towards Robot Skill Learning: From Simple Skills to Table Tennis*, in Machine Learning and Knowledge Discovery in Databases, Vol. 8190 of LNCS, Springer Berlin Heidelberg, pp. 627–631, 2013.

[17] A. L. Thomaz and C. Breazeal, *Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance*, in Proceedings of the 21st national conference on Artificial intelligence - Volume 1, ser. AAAI'06. AAAI Press, pp. 1000-1005, 2006.

[18] A. L. Thomaz, G. Hoffman, and C. Breazeal, *Real-Time Interactive Reinforcement Learning for Robots*, in Proceedings of AAAI Workshop on Human Comprehensible Machine Learning, 2005.

[19] W. B. Knox, P. Stone, and C. Breazeal, *Teaching Agents With Human Feedback: A Demonstration of the TAMER Framework*, in Proceedings of International Conference on Intelligent User Interfaces Companion, pp. 65–66, 2013.

[20] W. B. Knox and P. Stone, *Reinforcement Learning From Human Reward: Discounting in Episodic Tasks*, in Proceedings of RO-MAN 2012 IEEE, pp. 878–885, 2012.

[21] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. Thomaz, *Policy Shaping: Integrating Human Feedback With Reinforcement Learning*, in Advances in Neural Information Processing Systems, pp. 2625–2633, 2013.

[22] F. Cruz, S. Magg, C. Weber, and S. Wermter, *Improving Reinforcement Learning with Interactive Feedback and Affordances*, in Proceedings of the IV Joint IEEE ICDL-EpiRob, pp. 125–130, 2014.

[23] A. Chemero, *Radical embodied cognitive science*, MIT press, 2011.

[24] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, *To afford or not to afford: A new formalization of affordances toward Affordance-Based robot control*, in Adaptive Behavior, Vol. 15, no. 4, pp. 447–472, 2007.

[25] A. Chemero and M. T. Turvey, *Gibsonian affordances for roboticists*, in Adaptive Behavior, Vol. 15, no. 4, pp. 473–480, 2007.

[26] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, *Learning Object Affordances: From Sensory-Motor Coordination to Imitation* in IEEE Transactions on Robotics, Vol. 24, No. 1, pp. 15–26, 2008.

[27] İ. Atıl, N. Dağ, S. Kalkan, and E. Şahin, *Affordances and Emergence of Concepts*, in Proceedings of 10th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, pp. 149–156, 2010.

[28] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, *Your word is my command: Google search by voice: A case study*, in Advances in Speech Recognition, Springer US, pp. 61-90, 2010.

[29] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter, *Improving Domain-independent Cloud-based Speech Recognition with Domain-dependent Phonetic Post-processing*, in Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14), pp. 1529–1535, 2014.

[30] M. Bisani and H. Ney, *Joint-sequence models for grapheme-to-phoneme conversion*, Speech Communication, Vol. 50, no. 5, pp. 434-451, 2008.

[31] V. I. Levenshtein, *Binary codes capable of correcting deletions, insertions and reversals*, in Soviet Physics Doklady, Vol. 10, pp. 707, 1966.

[32] M. Kammer, T. Schack, M. Tscherepanow, and N. Yukie, *From Affordances to Situated Affordances in Robotics - Why Context is Important*, in Frontiers in Computational Neuroscience (Conference Abstract: IEEE ICDL-EPIROB 2011), Vol. 5(30), 2011.

[33] G. A. Rummery and M. Niranjan, *On-Line Q-Learning using Connectionist Systems*, in Cambridge University Engineering Department, 1994

[34] G. Cybenko, *Approximation by superpositions of a sigmoid function*, in Mathematics of Control, Signals, and Systems, Vol. 2, pp. 303–314, 1989.

[35] K. Funahashi, *On the approximate realization of continuous mappings by neural networks*, in Neural Networks, Vol. 2, pp. 183–192, 1989.

[36] K. Hornik, M. Stinchcombe, and H. White, *Multi-layer feedforward networks are universal approximators*, in Neural Networks, Vol. 2, pp. 359–366, 1989.

[37] Y. Bengio and O. Delalleau, *On the expressive power of deep architectures*, in Algorithmic Learning Theory, LNCS, Springer Berlin Heidelberg, Vol. 6925, pp. 18–36, 2011.

[38] J. Schmidhuber, *Deep learning in neural networks: An overview*, in Neural Networks, Vol. 61, pp. 85–117, 2014.

[39] K. P. Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012.

[40] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10), pp. 249–256, 2010.

[41] D. Nguyen, B. Widrow, *Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights*, in Proceedings IEEE IJCNN, pp. 21–26, 1990.

[42] M. T. Hagan and M. B. Menhaj, *Training feedforward networks with the Marquardt algorithm*, in Neural Networks, IEEE Transactions, Vol. 5, no. 6, pp. 989–993, 1994.

[43] C. Stahlhut, N. Navarro-Guerrero, C. Weber, and S. Wermter, *Interaction is more beneficial in complex reinforcement learning problems than in simple ones*, in Proceedings of the Interdisziplinärer Workshop Kognitive Systeme (KogSys), pp. 142–150, 2015.

(c) 2015 IEEE. Personal use of this material is permitted.