# Learning to Forage Through Imitation

Yiannis Gatsoulis George Maistros Yuval Marom Gillian Hayes
School of Artificial Intelligence
Division of Informatics
University of Edinburgh
5 Forrest Hill, Edinburgh, EH1 2QL
Scotland, UK
email: gatsoulis@hotmail.com

**ABSTRACT**
This paper reports on a set of learning experiments which could be considered as first steps towards learning at different levels of a hierarchical approach to imitation learning. A learner robot knows a priori how to execute some behaviours but does not know when. By following a teacher robot through a series of experiences and through imitating the demonstrated behaviours, the learner robot learns when to execute each behaviour and subsequently learns how to forage. The performance of the learner as well as the factors that affect its performance are described.

**KEY WORDS**
robotics, imitation learning, hierarchical imitation

## 1 Introduction

In the last decade roboticists have shown great interest in imitation learning. This interest arises from the fact that imitation learning is a powerful means of acquiring new skills within a social environment, found in humans and other animal species. It could lead into a more user friendly way of programming robots as a human could program a robot by just demonstrating the task.

Two forms of learning associated with imitation can be distinguished: *'learning to imitate'*, where the robot learns what to do with its motor system in order to perform the same behaviour as another robot; and *'learning through imitation'*, where "the robot learns by imitating the other agent and associating perceptual experiences with this motor act" [1]. Learning to imitate has been thoroughly studied in assembly robotic systems [2, 3] as well as in computer simulations and mobile robots [4, 5, 6]. There are also a few research attempts for learning through imitation in computer simulations [7] and in real robots [8, 9, 10].

In this study we investigate in real robots the efficiency and value of learning through imitation in a foraging scenario. A teacher performs foraging and leads a learner robot through a series of experiences, letting the learner know of the behaviours it executes. The learner robot follows the teacher, and by having the confidence that the teacher is doing the right thing it carries out the just told behaviours and associates them with its perceptions. Hence, we could say that the learner can in some sense imitate the teacher, even if it is not imitating the details of how to carry out the behaviour (it already knows) and it is not using its perceptions to tell it what behaviours to imitate (it is just doing what the teacher tells it). This imitation is necessary as the learner must get itself into the right sensory state for the next behaviour to be appropriate. The result of forming associations between perceptions and behaviours is that the learner learns to execute the behaviours in the right circumstances, and therefore becomes able to forage on its own.

## 2 Background

Imitation learning has been thoroughly described from an ethological perspective [11, 12], giving inspirations to roboticists for studying and applying imitation learning in robots.

A hierarchical approach of imitation learning was proposed after studies in animal behaviour [13]. In this study, imitation can occur in two distinct levels: " 'the action level', a rather detailed and linear specification of sequential acts, and the 'program level', a broader description of subroutine structure and the hierarchical layout of a behavioural program".

Our study comes close to the program level imitation. The long term objective is the implementation of a similar hierarchical architecture where an agent learns through imitation what behaviours to do (program level), and then again learn by imitation these behaviours individually (action level). The benefits are the same benefits of a modular approach, like any possible problems would be dealt 'locally', i.e. in the scope of a behaviour, rather than 'globally'.

One of the earliest studies of imitation learning in robotics involved a robot that was learning to negotiate a maze by following a teacher robot, which was an expert in navigating through a maze [14]. The passive and active imitation architectures were later introduced [15, 5]. In the former (passive imitation) the learner perceives the environment and the action demonstrated, recognises the demonstrated action and finally tries to imitate it, while in the latter (active imitation) the learner internally generates and tests candidate matching actions while the teacher demonstrates an action.

Imitation learning was also chosen as a staring point in studying artificial social intelligence and skills that are considered important for the development of social robots [16].

The transmission of a language from one robot to another using learning through imitation, such that the two robots acquire a common vocabulary describing their states and their actions is discussed in [8, 17]. Moreover, Robota, a robot doll able to learn, imitate and communicate, showed evidences of learning action sequences and words to describe these actions as well as its body parts in experiments that were conducted in the laboratory and by playing with 5–years old children [18].

In the case of two simulated humanoids, one of them is able to learn by imitation from the other repetitive patterns of arm and leg movements, oscillatory movements of the shoulders and the elbows, and precise movements for reaching and grasping imaginary objects using a neural architecture [19]. Furthermore, a simulated humanoid torso (named Adonis), showed the ability to learn how to dance the macarena from a verbal description of the dance movements [20, 4].

## 3 System Setup

Two of the well-known K-Team™Khepera™robots with their equipped gripper modules were used for the experiments. The training of the learner took place in two corridor arenas of approximately $230mm$ wide by $650mm$ long, one occupied by the teacher and the other by the learner. During the recall phase one of the training corridors was used for the first series of experiments, while a larger rectangular arena of $600mm$ in width and $650mm$ in length was used for the rest. An overhead camera above the arenas is used for tracking the two robots.

The overhead camera is one of the primary sensors of the robots. It provides the positions and the orientations of the robots as well as the positions of the food objects. Other sensors include the front infrared sensors, the optical barrier of the gripper module (implies if there is an object within the gripper), the position of the gripper motors (i.e. if the gripper is open or closed) and the height of the gripper arms from the ground.

## 4 The Foraging Task

In the following sections there are some terms that are frequently used and must be clarified. Firstly, the *raw data* has its usual meaning, as the sensor readings of the robot (of varying different modalities). We perform minimal processing to the raw data to form a prototype *state vector*, consisting of the object presence of the gripper module, the distance of the robot from the nearest food object, the distance of the robot from its home, the type of food and the height of its grippers from the ground. A state vector is an instance description of a state. Note that it is not necessary to know all the raw data in order for the robot to determine its state. The associations that the learner forms are exactly between these state vectors and the corresponding behaviours. We call these associations *experience vectors*. A file with such experience vectors is called *experience file*. The terms experience file and training episode are used interchangeably meaning the same thing, i.e. the experience gained after the training, even if the term training episode describes the training process as well.

The experimental scenario is described as a robot searches for food objects and carries them to its home. Starting from an initial position distant from the food and through the overhead camera it sees where the food lies and starts approaching it firstly at a fast speed and then slower in order to be more maneuverable. When it comes next to the food it starts orienting, and upon successful orientation the food type is identified. Depending on the food type the appropriate behaviour is executed. For instance, let's assume that the food can be lifted up; then the robot would pick up the object and it would carry it to its home. When the robot reaches its home it lets the food object down and leaves. The process is repeated until all food objects have been collected or until the episode terminates (i.e. after a certain number of steps).

Both robots perform foraging by executing the following routines in every step:
1. It perceives its environment.
2. The raw data is preprocessed to form a state vector.
3a. For the teacher, the state vector is classified into one of the possible states using a decision tree.
3b. For the learner, the state vector is best matched to one of the state vectors of the formed experience vectors from its training.
4. The corresponding behaviour is executed.

The foraging task has been broken down into 10 mutually exclusive states and 10 correct corresponding hand-coded behaviours, as they are shown in Table 1.

| **STATE** | **BEHAVIOUR** |
|---|---|
| SEARCH | SEARCH_FOR_FOOD |
| CLOSE_TO_FOOD | MOVE_SLOWLY |
| NEXT_TO_FOOD | ORIENT_WITH_FOOD |
| ORIENTED_WITH_FOOD | EXPLORE_FOOD |
| READY_TO_PICK_UP_FOOD | PICK_UP_FOOD |
| READY_TO_PUSH_FOOD | PUSH_FOOD |
| MUST_AVOID_FOOD | AVOID_FOOD |
| FOOD_IN_HAND | GO_HOME |
| AT_HOME_NO_FOOD | AVOID_HOME |
| AT_HOME_WITH_FOOD | RELEASE_FOOD |

Table 1. The states and their correct corresponding behaviours.

It appears from Table 1 that there is a logical sequence in the order that the states occur and hence also in the executed behaviours. It seems then that the robot will be able to forage successfully by following this order. However, this is not the case as the robot following the routines described above, only relies on its current perceptions to decide on which behaviour to execute, i.e. it only responds to its stimuli.

## 5 The Training Phase

Each robot (teacher and learner) occupies each own separate arena in the training phase. The two arenas are identical in respect to their dimensions, the initial positions of the robots, the positions of the food objects and the positions of the robot homes (Figure 1). The program consists of MultiSteps and Steps. A MultiStep is like a round in a board game while a Step is like a player's turn. Note that the teacher robot always takes its step before the learner.

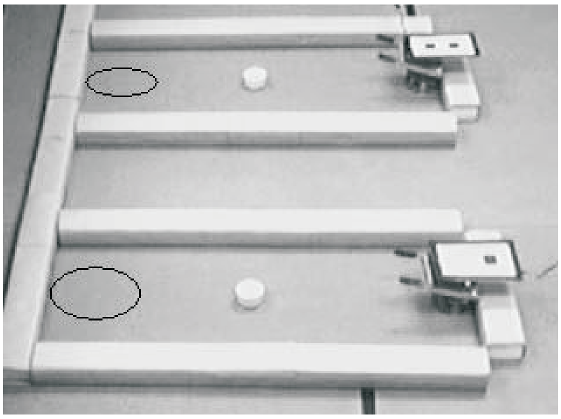

Figure 1. The training setup. The ellipses show the home areas.

The teacher on its step performs foraging as it would do if it were on its own except that it also 'shouts' the behaviour that executes.

Hence, at every step the teacher:
1. transforms the raw data into a state vector;
2. classifies the state vector into a state (using its decision tree);
3. 'shouts' the corresponding behaviour to the learner;
4. executes the behaviour.

Similarly, the learner at every step:
1. transforms its raw data into a state vector;
2. listens to the behaviour shouted by the teacher;
3. associates this behaviour with its state vector, forming an experience vector;
4. executes the behaviour;
5. and follows the teacher in an attempt to maintain identical spatial positions, if it is not already.

The learner, by recording its state vector and the corresponding behaviour (forming an *'experience vector'*), learns which behaviour should perform in a similar situation. This behaviour should be the correct one since the learner is in a similar situation (hopefully identical position) as the teacher who is a foraging expert. The learner then executes the behaviour it has just been told, resulting in imitating in some sense the teacher. *The learner robot has to carry out the behaviour at the time to get itself in the same perceptual state as the teacher for the next behaviour to be appropriate*. For example, if it did not imitate picking up the food, it would not know 'how it feels' to hold the food.

However, due to likely different configurations of the two robots and noise from the environment (like different frictions) the position of the learner may not be the same as the teacher's. In fact, it was observed that although the behaviour functions are the same, the two robots execute them slightly differently. For example, the two robots do not approach the food objects along the same path. This is why the last step the learner does is to follow the teacher in order to reach the same spatial position. However, this following does not guarantee that the learner would be in the same position with the teacher in the end of it, as it is just a small correction step towards the position and the orientation of the teacher. These corrections are so small that are not even visually observed.

For every training episode an experience file is produced. Note again that the terms training episode and experience file are used interchangeably. Hence, when the learner is further trained it means that it uses a combination (set) of these 'single' experience files.

The learner was trained ten times in the setup shown in Figure 1. Ten more training episodes were produced in a different setup, in which the positions of the robots and of the food objects have been switched from the ones in Figure 1. These last training episodes are incomplete as the training was stopped when the robots had picked up the food objects. These episodes are incomplete because we only wanted to further train the learner to this point, thus allowing us to save time. The only drawback of these incomplete training episodes is of course that they are not stand-alone, but they must be used in conjunction with the complete training episodes.

## 6 Experiments

In the recall phase the learner robot should be able to negotiate a foraging task in the absence of the teacher robot using the experience gained from its training.

The learner goes through the following processes at its every step:
1. It perceives the environment.
2. It transforms the raw data into a state vector, called input state vector.
3. The experience vectors with the most similar state vectors to the input state vector are then found.
4. The associated behaviour of these experience vectors is finally executed.

The experience vectors with the most similar state vectors to the input state vector are found by using a simple nearest–neighbour algorithm. It is likely that the set of the most similar experience vectors do not all correspond to the same behaviour. In this case the behaviour which is executed is selected randomly with a probability that corresponds to the size of each subset of the candidate behaviours. This probability is called 'confidence level', based on our confidence that correct experience vectors should dominate over the noisy vectors as more experience is acquired.

The learner is given a limited amount of time (i.e. steps) to collect the food objects, based upon the average number of steps needed for the teacher to clear the arena.

When all the food objects are carried home or its time runs out the evaluation episode ends. The evaluation measurement is a score calculated by:

- Counting the number of 'correct' behaviours (i.e. the ones that the teacher would execute) and normalising them into a percentage by dividing with the total number of executed behaviours (steps).

- A penalty is then applied, depending on how many food objects the learner failed to collect and are still lying in the arena.

Hence, the score calculation can be expressed with the following equation:

$$Score = \frac{C}{T} - P, \tag{1}$$

where C: number of correct behaviours, T: total number of steps, P: penalty for uncollected food.

The number of correct behaviours of the learner is a measurement of how well it has learned to forage. However, this is not an objective criterion as it does not capture the whole scope of the foraging process. For instance, the learner could spend most of its time 'correctly' searching for food. Therefore, the learner would achieve a high score of correct behaviours without bringing any food to its home or doing anything other than searching. This is why a penalty is given, which depends on the number of the uncollected food objects.

*In the first series of experiments* the performance of the learner was evaluated in one of the training corridors. Due to the small size of the arena only one available food object was used. The time that the learner had to clear the arena was 500 steps. The penalty for failing to collect the food was -20. The initial position of the learner and the position of the food were random in every recall episode. The learner was asked to forage (recall episode) with the experience of 1, 3, 5 and 10 training episodes (experience files). In each case there were 20 recall episodes. In these 20 recall episodes of every case we used 5 random sets (combinations of experience files) from the possible available ones, each one running it 4 times. For example, for 3 training episodes we used 5 random triplets of experience files from the possible available ones, each one running it 4 times ($5x4 = 20$ recall episodes for 3 experience files). Figure 2 shows the average scores (and the number of collected food objects) with respect to the number of training episodes used. The average score for 1 training episode is 53, which increases quite significantly to 68 for 3 training episodes and reaches a maximum value of 74 for 10 training episodes. The percentage of collected food objects is also nearly doubled from 40% (8/20) when using 1 training episode to 70% (14/20) when using 10.

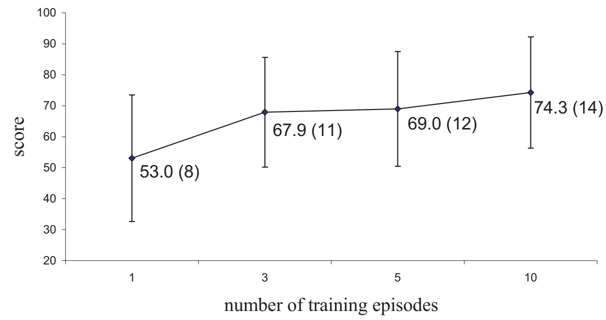


Figure 2. Graph showing the average scores with respect to the number of training episodes in the training corridor. The numbers in the parentheses are the collected food objects (possible maximum 20). The error bars are the standard deviations.

*In the second series of experiments* a larger rectangular arena was used with two available food objects. The time limit was set to 700 steps. The penalties were -20 for failing to collect any food (either of the two), and -5 for failing to collect the second food object (i.e. collect one but not the other). The initial positions of the learner and of the food were random. Similar to the first set of experiments the learner was asked to forage with the experience of 1, 5 and 10 training episodes. For each one there were 12 recall episodes. The number of random sets in each case were 3, running each one of them 4 times. The average scores (and the number of collected food objects) with respect to the number of training episodes are shown in Figure 3. For 1 training episode the score was 45 and increases quite significantly to 61 for 5 training episodes and to 68 for 10. The percentages of the collected food objects were 30% (7/24), 58% (14/24) and 54% (13/24) respectively.

*In the third series of experiments* the same rectangular arena was used. The setup was similar to the setup of the second series of the experiments except from the time limit to clear the arena which was increased to 800. As a result the penalty was -10 per uncollected food. Furthermore, sets of 1, 3, 5, 10, 15 and 20 training episodes were used. The number of recall episodes in each case was 25. In these experiments all the single training episodes were filtered so as to remove the experience vectors with the same state vectors. The sets of 1, 3, etc. training episodes were formed from these filtered ones. The reason for this filtering is discussed in the next section. The average scores (and the

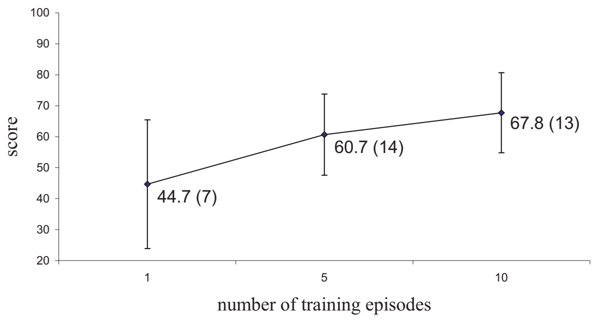


Figure 3. Graph showing the average scores with respect to the number of training episodes in the rectangular arena for the second series of experiments. The numbers in the parentheses are the collected food objects (possible maximum 24). The error bars are the standard deviations.

number of collected food objects) with respect to the number of training episodes are shown in Figure 4. For 1 training episode the score was 69. The number of collected food objects was 25 out of the 50 possible ones, which gives a rate of 50%. The performance steadily increases as more training episodes are used reaching the highest value of 88, and having collected 92% (46/50) of the food objects with 20 training episodes. The second and the third best results are also quite close to the best ones. With 15 training episodes the score was 87 with 88% (44/50) of the food objects collected. The score was 85 and having collected 78% (39/50) of the food objects with 10 training episodes, i.e. half the training episodes used for the best results.

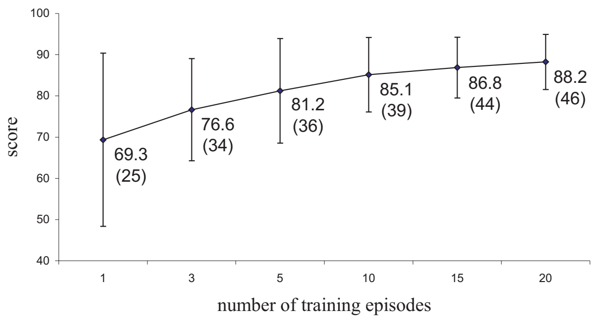


Figure 4. Graph showing the average scores with respect to the number of training episodes in the rectangular arena for the third series of experiments. The numbers in the parentheses are the collected food objects (possible maximum 50). The error bars are the standard deviations.

*In the fourth series of experiments* the setup was the same as the setup of the third series. Sets of 1, 10 and 20 training episodes were used. The number of recall episodes in each case was 25. The sets were formed using the filtered experience files. However, the prototype of the state vector of the experience vectors was changed. In the first series of experiments the distance of the robot from its home and the distance of the robot from the nearest food were Euclidean distances. This time they were changed into boolean type (at home or not at home) and into relative distance (far from food, close to food and next to food) respectively. The reasoning is explained in the discussion section. Figure 5 shows that the score was 68 and the collected food objects were 38% (19/50) for 1 training episode. These numbers improve until a certain number of training episodes. For 10 training episodes the score was 73 and collected 68% (34/50) of the food objects. The results when using 20 training episodes are similar to the ones of 10 training episodes. The score was 73 and the learner collected 78% (39/50) of the food objects.

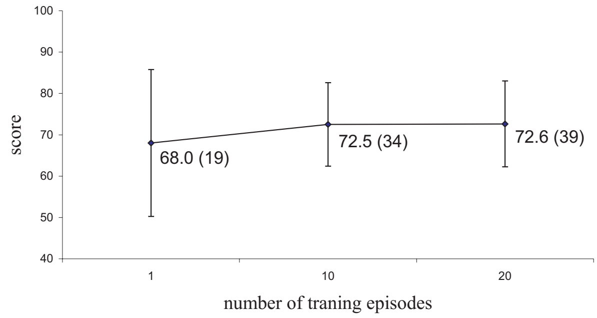


Figure 5. Graph showing the average scores with respect to the number of training episodes in the rectangular arena for the fourth series of experiments. The numbers in the parentheses are the collected food objects (possible maximum 50). The error bars are the standard deviations.

## 7 Discussion

The experiments showed that the learner can generalise beyond its training environment, in arenas though that are still very similar to the training one.

Furthermore, all the experiments showed that the performance of the learner increases as more experience is acquired. However, this improvement is more significant in the early stages.

Two difficulties were observed, particularly in the first two experiments. Firstly, the learner had difficulty in executing the behaviour of orienting with the food when it was next to it. Instead, it was approaching it and thus avoiding it as an obstacle quite frequently. The reason lies in the method used for deciding which experience vector to choose when there were more than one behaviours that could be executed. Due to the slow update of the camera sensor, which perceives the distance of the robot from the food object, it was very likely that there were more experience vectors with the approach food behaviour than with the orient behaviour. Hence, the confidence (probability) of choosing to approach the food was higher. Since this was happening in most of the experience files, the noisy experience vectors were dominating over the correct ones even when more experience was acquired, and as a result the problem still remained. It is worth mentioning that everything was executed correctly as long as the learner oriented successfully. In the third and the fourth experiments the single experience files were filtered, leaving only unique experience vectors. Furthermore, the learner was further trained paying attention at this difficult point without needing complete training episodes. This filtering and the extra experience reduced the problem down to an expected and normal probability that the noisy vectors should have.

Another problem that was visually observed in the first two experiments was that the learner was executing the behaviour of orienting with food in a particular area of the arena, even if there was not any food there, as its input state vector was classified to the experience vector with the orient behaviour. This was a result of using training episodes that were produced from the same training setup, which did not include experiences similar to the one it was facing in its recall phase. In the third and fourth experiments the extra experience of the learner was from a different training setup (the incomplete training episodes) as well, and the problem was completely resolved.

The best performance of the learner was achieved in the third series of experiments. The learner with 20 training episodes achieved a 88 score. The reliability of performance is shown from the low standard deviation as well as from the fact of managing to collect 46 food objects out of the possible 50 (92%). It also managed to outperform the teacher in the average number of steps, 504 steps for the learner in comparison to the 540 steps for the teacher. In an attempt to improve the performance of the learner the prototype of the state vector was changed. The distances of the learner from its home and from the nearest food were transformed from Euclidean distances into boolean and enumerated types respectively. The reasoning was that boolean and enumerated types are considered to be more general descriptions than integers. However, as a result of this generalisation the population of the noisy vectors increased, unlike the correct ones. This can be verified by the worse results of the experiments.

The second and the third best results were also achieved in the third series of experiments when using 15 and 10 training episodes respectively. For 15 training episodes the score was 87, with a similarly low standard deviation and with a number of collected food objects rising to 44 out of 50 (88%). For 10 training episodes the score was 85, with a slightly higher standard deviation and it managed to collect 39 out of 50 (78%) food objects. The results of these two cases are quite close to the best results obtained for 20 training episodes. The usual trade–off of further training (time consuming) versus performance increases, as it seems that the improvement of the performance is minor for very extensive training. In our case, where the scenario is not very complex and the number of states and behaviours is small, the time spent for further training was not much. However, in cases where the task might be much more complicated and the time spent for training is vital, a solution close to the best one, but with likely considerable time savings, could possibly be preferred.

## 8 Conclusions

In this study we describe on a set of learning through imitation experiments, based on a hierarchical approach to imitation learning.

These experiments showed that the learner can generalise beyond its training arena. This means that the training can take place in an environment with facilitation that make learning easier, and still the robot will be able to cope with any arena of the same kind (same food objects, similar home, etc.). However, more work needs to be done to make it properly generalise, since the two arenas are still very similar.

Furthermore, the performance of the learner indeed increases as more experience is acquired. However, it seems that this improvement is more significant in the early stages. Moreover, if there is a suboptimal solution close to the best one when using less experience, then the issue of a tradeoff between the time spent for the training and the efficiency of the learner is considered, i.e. the question of how much training is needed before a satisfactory solution is achieved considering the cost of the time spent is asked.

The performance of the learner is affected by some other factors as well. One of them is the 'spatial variety' of the training episodes. The problems of executing wrong behaviours at certain perceptions were resolved with more experience that was acquired from different training setups.

The filtering of the single training episodes into unique experience vectors seems to significantly improve the performance of the learner. Hence, when we say that the teacher leads the learner through a series of experiences, we should talk about *unique experiences*.

Finally, the performance of the learner is also affected by the design of the state vector prototype. It can be designed to be similar or even the same as the one of the teacher robot when the two robots are identical, and yield good results. Naturally, it is unlikely to be the case if the morphologies of the teacher and the learner are different.

In further work, the learner should also be able to recognise the behaviour of the teacher from its own perceptions, i.e. it has to be able to recognise that the teacher is "grasping the food". So a complete system would:

1. Learn the details of how to do each behaviour (low level), including start and end points.

2. Learn when to do each behaviour (high level).

3. Recognise from observing the teacher's actions/movements what behaviour the teacher is actually doing at the moment.

These experiments indicate that a hierarchical approach to imitation learning lends encouragement to the continuation of this approach. However, it cannot be fully evaluated until a complete imitation system is implemented.

## Acknowledgements

Facilities for this work were provided by the University of Edinburgh. Yiannis Gatsoulis was funded by an EPSRC studentship (award number: 00401579).

## References

[1] J. Demiris and G. Hayes. Do robots ape? In *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, pages 28–31, Tang Center - MIT, Cambridge, Mass., November 1997.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[3] S. Schaal. Learning from demonstration. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 1040–1046, Cambridge, MA, 1997. MIT Press.

[4] M. Mataric. Getting humanoids to move and imitate. *IEEE Intelligent Systems*, pages 18–24, July 2000.

[5] J. Demiris and G. Hayes. Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In K. Dautenhahn and C. Nehaniv, editors, *Imitation in Animals and Artifacts*, chapter 13. MIT Press, 2001.

[6] A. Alissandrakis, C.L. Nehaniv, and K. Dautehahn. Learning how to do things with imitation. In *AAAI Fall Symposium Series*, Sea Crest Oceanfront Resort and Conference Centre, North Falmouth, Massachussets, USA, November 2000.

[7] G. Maistros and G. Hayes. An imitation mechanism for goal-directed actions. In *Proceedings TIMR 2001 - Towards Intelligent Mobile Robots*, Manchester, April 2001.

[8] A. Billard and K. Dautenhahn. Grounding communication in situated, social robots. In *Proceedings of the Towards Intelligent Mobile Robots Conference*, Manchester, 1997.

[9] A. Billard and G. Hayes. Transmitting communication skills through imitation in autonomous robots. In A. Birk and J. Demiris, editors, *Learning Robots: A Multi-Perspective Exploration*, Lecture Notes in Artificial Intelligence. Springer–Verlag, Brighton, UK, 1998.

[10] Y. Marom and G. Hayes. Attention and social situatedness for skill acquisition. In *Proceedings of the First International Workshop on Epigenetic Robotics: Modelling Cognitive Development in Robotic Systems*, Lund, Sweden, September 2001.

[11] B.G. Jr. Galef. Imitation in animals: History, definition, and interpretation of data from the psychological laboratory. In *Social Learning: Psychological and Biological Perspectives*, pages 3–28, Hillsdale, N.J., 1988. Lawrence Erlbaum Associates.

[12] A. Miklosi. The ethological analysis of imitation. *Biological Review*, 74:347–374, 1999.

[13] R.W. Byrne and A.E. Russon. Learning by imitation: a hierarchical approach. *Brain and Behavioral Sciences*, 21(5):667–721, 1998.

[14] G. Hayes and J. Demiris. Robot controller using learning by imitation. In *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, pages 198–204, 1994.

[15] J. Demiris and G. Hayes. Active and passive routes to imitation. In *Proceedings of the AISB Symposium on imitation on animals and artifacts*, Edinburgh, Scotland, UK, 1999.

[16] K. Dautenhahn. Getting to know each other – artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356, 1995.

[17] A. Billard and G. Hayes. DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1):35–64, 1999.

[18] A. Billard, K. Dautenhahn, and G. Hayes. Experiments on human–robot communication with Robota, an imitative learning and communicating doll robot. Technical Report CPM-98-38, Centre for Policy Modelling, Manchester Metropolital University, 1998. Contribution to Workshop Socially Situated Intelligence at SAB98 conference, Zurich.

[19] A. Billard and M.J. Mataric. A biologically inspired robotic model for learning by imitation. *Cybernetics and Systems Journal, special issue on Imitation in animals and artifacts*, pages 373–380, 2000.

[20] M. Mataric, V.B. Zordan, and Z. Mason. Movement control methods for complex, dynamically simulated agents: Adonis dances the macarena. In *Proceedings of Autonomous Agents*, pages 317–324, Minneapolis/St. Paul, May 1998. ACM Press.