

Do as I Do: Correspondences across Different Robotic Embodiments

Aris Alissandrakis, Chrystopher L. Nehaniv, and Kerstin Dautenhahn

Adaptive Systems Research Group
University of Hertfordshire
Hatfield, Herts AL10 9AB, United Kingdom

{A.Alissandrakis, C.L.Nehaniv, K.Dautenhahn}@herts.ac.uk
<http://homepages.feis.herts.ac.uk/~nehaniv/ASRG>

Abstract

Behaviour matching and imitation serve as fundamental mechanisms for social learning, the development of social skills, and the evolution of cultures. Imitation and observational learning as means for acquiring new behaviours also represent a largely untapped resource for robotics and artificial life - both in the study of “*life as it could be*” and for applications of biological mechanisms to synthetic worlds. A crucial problem in imitation is the *correspondence problem*, mapping action sequences of the demonstrator and the imitator agent. This problem becomes particularly obvious when the two agents do not share the same embodiment and affordances. This paper describes work-in-progress using the general imitating mechanism ALICE (Action Learning for Imitation via Correspondences between Embodiments), trying to find solutions for the correspondence problem between different configurations of robotic arms.

1 Introduction

A characteristic of many social animal species *as-we-know-them*, e.g. dolphins, chimpanzees, humans and other apes, is the ability to learn from others by imitation. Inspired by nature, over the past decade many researchers have attempted to design life-like social artifacts *as-they-could-be*, i.e. software or robotic artifacts that are able to learn from each other or from human beings. A human that can teach a robot or a software program simply by showing or demonstrating what needs to be done is an exciting new programming paradigm (Cypher 1993, Atkeson, Hale, Pollick, Riley, Schaal, Shibata, Tevatia, Ude, Vijayakumar, Kawato & Kawato 2000).

While research on imitation usually takes the approach of studying *learning by imitation* (assuming that an artifact already possesses the skill to imitate successfully), this paper addresses the complementary approach of *trying to imitate* or *learning how to imitate* (Dautenhahn 1994). Unlike other work on imitation, which assumes that demonstrator and imitator share the same embodiment and provide an *ad hoc* mapping between them, our work systematically investigates constructive solutions of the *correspondence problem* between dissimilar embodiments.

2 The Importance of Embodiment in Imitation and the Correspondence Problem

Imitation is a scientific challenge in many ways. In addition to fundamental problems of who, what, when and how to imitate, finding mappings between corresponding body parts and actions (e.g. lifting the right leg when imitating another human who is lifting the right leg) is a major challenge for artifacts to be solved. The work presented in this paper specifically addresses this *correspondence problem* (Nehaniv & Dautenhahn 2000, Alissandrakis, Nehaniv & Dautenhahn 2000, Dautenhahn & Nehaniv 2002a).

Once an agent has decided who, when and what to imitate, the appropriate mechanisms must be employed to achieve the necessary imitating actions. The embodiment of the agent and its affordances also play a crucial role, as stated in the following *informal description of the correspondence problem*:

Given an observed behaviour of the model, which from a given starting state leads the model through a sequence (or hierarchy) of sub-goals in states, action and/or effects, one must find and execute a sequence of actions using one's own (possibly dissimilar) embodiment, which from a corresponding starting state, leads through corresponding sub-goals - in corresponding states, actions, and/or effects, while possibly responding to corresponding events.

See (Nehaniv & Dautenhahn 2000, Nehaniv & Dautenhahn 2001a, Nehaniv & Dautenhahn 2002).

The agents may not necessarily share the same morphology or may not have the same affordances even among members of the same “species”. This is true for both biological (e.g. differences in height among humans) and artificial (e.g. differences in motor and actuator properties) agents. Having similar embodiments and/or affordances is just a special case of the more general problem. Differences in embodiment between animals, robotic and software systems make it more difficult, but not necessarily impossible to acquire corresponding behaviours.

In our approach, a correspondence is a “recipe” through which an imitator can map observed actions of the demonstrator to its own repertoire of actions as constrained by its embodiment and by context (Nehaniv & Dautenhahn 2000, Nehaniv & Dautenhahn 2001a, Nehaniv & Dautenhahn 2002). A correspondence thus serves as a ‘looking-glass’ through which an observed demonstrator’s behaviour is ‘refracted’ to yield similar, but possibly not quite the same, action sequences for the imitator. This allows it to get along in its environment, using the affordances of its own embodiment, while exploiting observations of the behaviour of others.

3 Travelling Through the Looking-Glass with ALICE

A mechanism that solves the corresponding problem is an essential part of any artificial system with successful imitating behaviour capabilities. In (Alissandrakis, Nehaniv & Dautenhahn 2001) we introduced ALICE (Action Learning for Imitation via Correspondences between Embodiments) as a generic mechanism for building up a correspondence, based on any imitation action generating method, by examining the history of such imitation attempts (cf. Byrne’s string parsing approach to imitation in animals (Byrne 1999)). This correspondence library that ALICE builds up can then be employed when imitating. For examples of biological and artificial agents solving the

correspondence problem see, e.g. the imitation of humans by dolphins (Herman 2002), of human referential speech by parrots (Pepperberg 2002), and other articles in (Dautenhahn & Nehaniv 2002b, Nehaniv & Dautenhahn 2001b). (Nehaniv & Dautenhahn 2000, Nehaniv & Dautenhahn 2001a)).

ALICE is comprised of two major components on top of any generating mechanism of imitation attempts. First, when the imitator observes a new demonstrator action not seen before, the imitator can relate the attempted match result (given by the generating mechanism used) to that demonstrator action. The generating mechanism result might be one, or a sequence of more than one imitator actions. This relation of observed demonstrator actions to attempted matching sequence of imitator actions is then placed as a new entry in the library of correspondences. At each stage in its growth, a library of correspondences is an example of a (partial) relational homomorphism between the abstract automata associated to the demonstrator and the imitator (Nehaniv & Dautenhahn 2001a, Nehaniv & Dautenhahn 2002).

Together with the pairs relating demonstrator actions to imitator sequences of actions in the correspondence library we also store the *degree of success* of these attempted matching behaviours. The degree of success is given by a *metric* which the imitator uses to *evaluate its own success*. Using different metrics determines *what* is being imitated, i.e. which aspects of the observed behaviour the imitator is required to match. As we have shown in previous work (Alissandrakis et al. 2000, Alissandrakis et al. 2001), the choice of metric can radically alter the character of the imitative behaviour.

Using the entries in the library instead of performing the algorithm for actions already observed is less computationally expensive; especially when the complexity of the algorithm that produces the matching behaviour increases. The cost of recalculating instead of using an already found solution may be considerable - for example a ten degrees of freedom robot arm controller having to solve again and again the same inverse kinematics equations.

If the correspondence library entries depend only on the generating mechanism used, any limitations of this mechanism would reflect on the quality and performance correspondences. Also, some of the found imitator sequences related to demonstrator actions may even become invalid in certain contexts. Therefore it is important to also keep track of alternative actions or actions sequences in the correspondence library. (For examples of such context effects and overcoming them with alternatives, see (Alissandrakis et al. 2001)).

The second component of ALICE addresses this issue: To discover further novel sequences the imitator agent can *examine its own history* without having to modify or improve the generating algorithm used. By history we mean the sequential list of actions that were performed so far by the agent while imitating the demonstrator together with these actions' resulting change of the agent's state and possible effects on the environment. This history provides helpful experience data that ALICE uses to extract useful mappings to improve and add to the correspondence relation library created up to that point.

A summary of the ALICE mechanism is given by the following pseudocode.

ALICE MECHANISM PSEUDOCODE

<i>Component #1</i>	<i>Component #2</i>
Consider the current demonstrator behavior as a sequence of actions.	In parallel, examine continuous sequences of past actions performed so far from history.
For each of these demonstrator actions either:	For each of these sequences:
<ul style="list-style-type: none">• If the demonstrator action has not been observed before, create new entry in the correspondence library and add sequence of imitator actions found by the generating mechanism.• If entry already exists, use appropriate imitator actions sequence from the correspondence library.	<ul style="list-style-type: none">• If sequence of imitator actions produces same effects to known demonstrator action, add sequence to that entry in the correspondence library.

The methods for extracting this information from history can vary, and managing the found sequences can depend on additional metrics, e.g. keeping only the shortest sequence that can achieve that change of state/effect, or keeping only the top five sequences according to a performance measure. The effective size of the history can depend also on the actual implementation and/or context.

4 Implementing the ALICE Mechanism in a Robotic Arm Test-bed

In previous work, the *chessworld* scenario (Alissandrakis et al. 2000, Alissandrakis et al. 2001) was used to study the correspondence problem. In the chessworld scenario we studied how chess pieces with dissimilar embodiments (and different movement repertoires) could learn to imitate each other. At the current stage of our research, the ALICE mechanism is now implemented on a different test-bed. The Swarm simulation system was used to create software agents embodied as robotic arms, each operating in a separate two-dimensional workspace. The model is simplified: it does not include physics or motor and actuator controllers – the transition from one state to the next is instantaneous. This allows us to experiment with a variety of polar configurations (with different number of joints n , each of them of length l_i , ($1 \leq i \leq n$)). Each different configuration between demonstrator-imitator pairs is effectively a dissimilar embodiment, allowing various degrees of imitation success. The ALICE mechanism is used to solve this correspondence problem. See Figure 1.

Each demonstrating agent can perform a model behaviour that consists of moving around in the workspace. As the arm moves around, several different features of that behaviour could be picked up and selected for imitation: either the trajectory of the end point, the trajectories of each joint

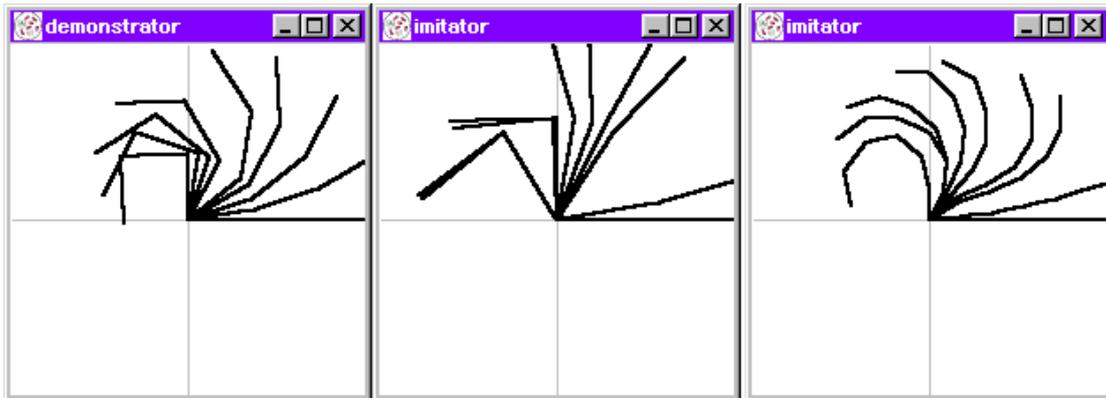


Figure 1: **Solving the correspondence problem with ALICE.** The demonstrator model behaviour consists of folding its 3 joints counter-clockwise (left). Intermediate stages of trajectories in imitative attempts to match the position of the end point using ALICE together with the random generating mechanism are shown for a 2-joint (centre) and a 6-joint (right) imitator. For the examples shown, the overall length of the robot arm remains constant, independent of the number of joints.

combined, the workspace area that was covered, or perhaps the value of the angles for each joint (or even for some particular joint). Furthermore if any objects were available within the workspace, or the arm was able to use tools, different kinds of effects on the environment could also be considered, besides the changes of the robot state (e.g. pick up boxes, leave trails of paint using a brush). At the current stage there are no objects in the workspace, and the manipulator can only move around. In the examples here, the state of the robotic arm is expressed as a vector $\bar{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$, where n is the number of joints in the arm configuration.

The process of segmenting the demonstrator behaviour into a sequence of actions to be imitated involves deep issues of perception that are simplified in our model. We assume that the system has an appropriate perceptual mechanism that takes care of that part of the process, and we concentrate instead on how to use the resulting data to reconstruct the desired result. At the current stage, any variation of less than ten degrees in the angle values for the joints is ignored in perception. Note that both the demonstrator and the imitator agents are still able to rotate their joints by an amount of less than ten degrees.

In previous work with the chessworld scenario we studied different metrics. At the current stage, only a simple metric is used to evaluate the success of the imitation: assuming the demonstrator and the imitator occupy the same workspace, measure the distance between their arm end points during the performance of the model behaviour. For a close imitation fit, this distance should be minimised. This metric can be applied even though the number of degrees of freedom or joint lengths may differ between the demonstrator and imitators. Therefore we implement a very simple greedy-type imitating algorithm. Instead of a generating mechanism that employs inverse kinematics to suggest appropriate values, an increase or decrease by a random amount of up to ten degrees for each of the components θ_i of vector $\bar{\theta}$ is used as the attempted matching solution. This solution is then evaluated using forward kinematics, leading to a position in the workspace for the end point of the imitator. Ideally for a successful imitation the Euclidean distance between this position and the demonstrator's position should be zero but the dissimilar embodiments, and of course the random nature of the attempted solution, make that unlikely. Nevertheless this consti-

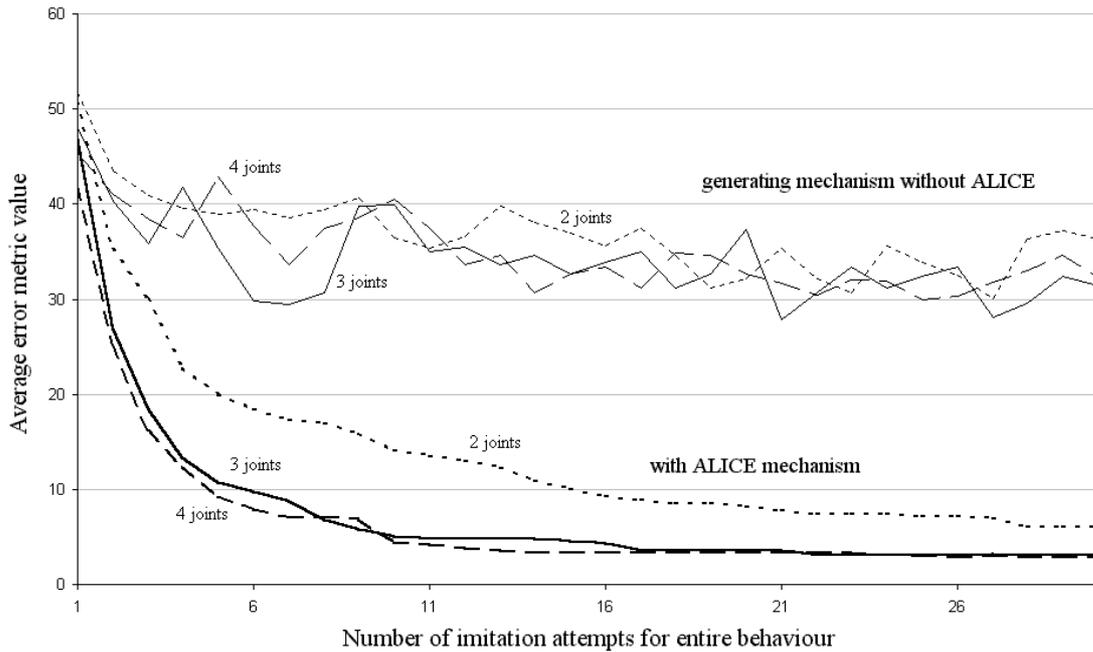


Figure 2: **Evaluating imitative performance.** The imitator repeatedly attempts to perform the 3-jointed demonstrator’s model behaviour (shown above in Fig. 1) using as a self-evaluation metric the distance between the end points of the demonstrator and imitator arms. This error (shown averaged over the duration of each attempt) is minimised when the ALICE mechanism is used (thicker lines), compared to when the generating mechanism is used on its own without ALICE (thinner lines). This indicates better performance, as the trajectory of the end point is more faithful to the desired one. The plots for three different imitator embodiments are shown, one with 2 joints (dotted lines), one with 3 joints like the imitator (solid lines) and another with 4 joints (dashed lines).

tutes a valid (but unsuccessful) imitation attempt. This metric is used both for self-evaluation by the imitator on the similarity of its actions to the desired ones, and also to measure the imitation performance as seen in Figure 2.

An obvious way to improve the imitation performance is to replace our random generating algorithm with one that employs inverse kinematics. According to the complexity of the imitator’s configuration, that might have a significant computational cost. Instead we can keep the existing algorithm and augment its performance by adding the ALICE mechanism. (See Figure 3.)

Using the random solutions as the imitator makes several attempts to produce the desired behaviour, ALICE creates a correspondence library. Each entry relating to each of the observed demonstrator states is corresponding to an imitator state found by the generating mechanism. Initially these correspondences will have very poor imitation results, but as the generating algorithm produces more random solutions, they can be replaced/updated by possibly more appropriate ones. Each entry in the correspondence library is rated according to the metric used, which relates to the aspect of the behaviour that is to be imitated. The metric used in these examples (distance between end-points) is a simple one, but more complex ones can be used, that also include kinematics and dynamics restrictions to make the solutions more relevant to the real world.

Related to this work is the Associative Sequence Learning (ASL) theory (Heyes & Ray 2000,

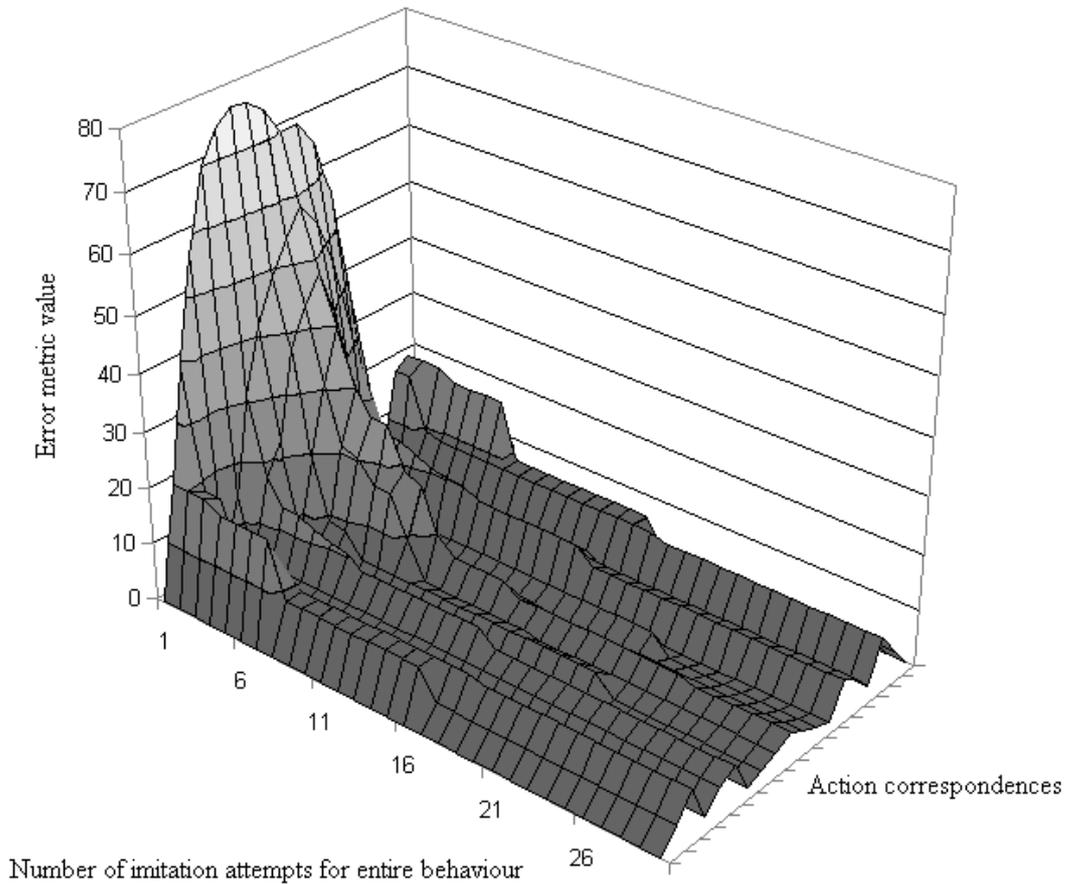


Figure 3: **Learning action correspondences.** For the demonstrator model behaviour shown in Fig. 1 above, ten entries in the correspondence library are created (a_0, \dots, a_9), each a_n corresponding to a demonstrator state $\bar{\theta} = [10n, 10n, 10n]$. Here imitator and demonstrator have the same 3-jointed type of embodiment. At each entire behaviour demonstration the imitator observes the actions a_0 through a_9 (folding counter clockwise) and then a_9 down to a_0 (for the demonstrator arm to return to its initial position). As the imitator makes attempts to reproduce the behaviour, the value of the error metric (distance between demonstrator-imitator end-points) is seen to decrease for each demonstrator action a_n as a result of the learning process. The figure shows that the most relatively difficult actions to learn correspond to the part of the behaviour when the arm is the most curled-up.

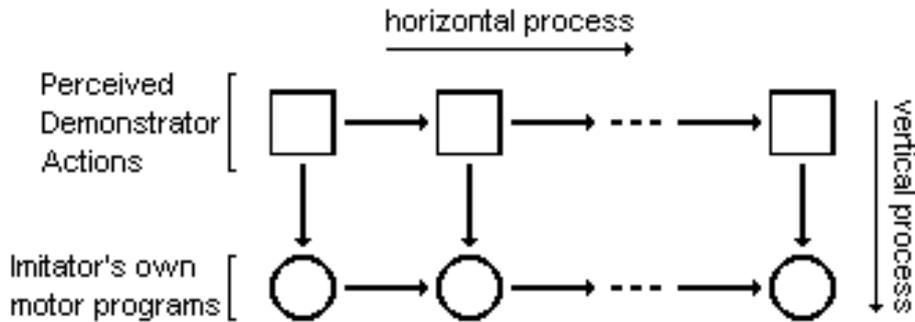


Figure 4: Associative Sequence Learning (ASL) theory requires two processes to successfully take place in imitation. The demonstrated behaviour is broken down into a sequence of elementary action units. The horizontal process associates the mental representations of the action units in a successive chain. The vertical process associates directly or indirectly each of the sensory representations to appropriate motor representations. Once both learning processes are complete, the imitator knows what the sequence “looks like” and how to perform that sequence in temporal order with appropriate motor programs for its own embodiment.

Heyes 2002), a proposed theory of imitation from psychology that makes testable predictions (see Fig. 4). In our work the correspondence library captures the vertical associations, while the temporal sequence of demonstrator actions relates to the horizontal process of ASL. However, one of the weak points of ASL is that it does not address effects on the environment, however our framework allows this to be handled using appropriate metrics.

5 Conclusions

Imitation and behavioural matching can serve as fundamental components for behaviour acquisition. This paper describes work-in-progress of the latest implementation for the ALICE (Action Learning for Imitation via Correspondences between Embodiments) generic imitation mechanism. The experimental test-bed consists of a simple software simulation of robotic arm polar configurations (moving in a two dimensional workspace) able to have a different number of joints n of length l_i ($1 \leq i \leq n$). Several different demonstrator-imitator pairs with dissimilar embodiments are used to study the correspondence problem using the ALICE mechanism. This paper presents some early results from work-in-progress involving robotic manipulators.

In the examples presented in this paper, the particular generating mechanism that was used could produce very poor results on its own, but performed successfully when augmented with ALICE. This success was achieved even when the imitator and demonstrator were robotic arm agents that

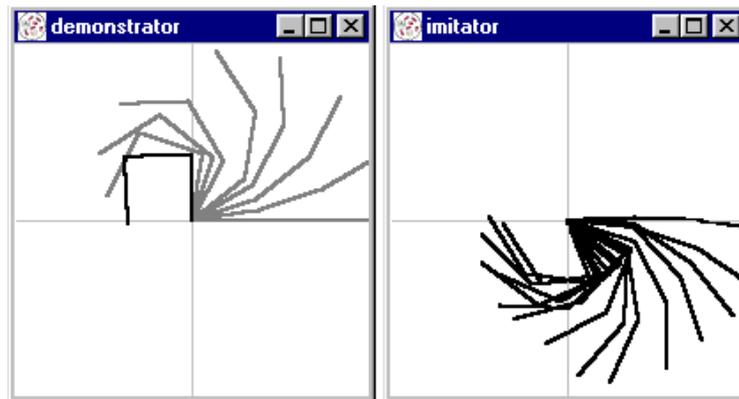


Figure 5: **An example of considering a metric with different granularity.** If instead of the entire demonstrator model behaviour shown in Fig. 1 above, the imitator measures its success considering only the final stage (left, all joints folded counter-clockwise) the imitator is able to find this unexpected – yet valid (according to the end-point distance metric) – corresponding solution (right, all joints folded clockwise).

had different numbers of degrees of freedom in the number of joints of their respective embodiments. Of course, differences and limitations of embodiment may lead to differing success at imitating the demonstrator's behaviour according to a given metric for success. For example, we saw somewhat less success for 2-jointed than 3- and 4-jointed imitators of a 3-joint demonstrator, while the latter two imitators exhibited about equal success according to the metric.

The generic nature of the ALICE mechanism allows for any generating mechanism, including more complex ones with better performance. But the real benefit of using the ALICE mechanism is having a correspondence library. On-line, the re-use of already found solutions can potentially minimise time and save computational resources substantially. These correspondences can be further improved if better ones are discovered (evaluated according to a metric). Off-line these correspondences provide useful data that can be studied in the case of discovering unexpected but valid behaviours (see Figure 5 for an example).

The ALICE mechanism is a generic learning and memory augmentation to any imitating system. Future work will address systematic experimental studies using this robotic arms test-bed, as well as study examples of more complex imitative behaviour. Studying effects on the environment is a particularly interesting line of future research.

References

- Alissandrakis, A., Nehaniv, C. L. & Dautenhahn, K. (2000), Learning How to Do Things with Imitation, *in* 'Proc. 2000 AAAI Fall Symposium on Learning How to Do Things', American Association for Artificial Intelligence, pp. 1–6.
- Alissandrakis, A., Nehaniv, C. L. & Dautenhahn, K. (2001), Through the Looking-Glass with ALICE– Trying to Imitate using Correspondences, *in* 'Proc. First International Workshop on Epigenetic Robotics: Modelling Cognitive Development in Robotic Systems', Lund University Cognitive Studies Series, Vol. 85, pp. 115–122.

- Atkeson, C. G., Hale, J. G., Pollick, F., Riley, M., Schaal, S. K. S., Shibata, T., Tevatia, G., Ude, A., Vijayakumar, S., Kawato, E. & Kawato, M. (2000), 'Using Humanoid Robots to Study Human Behavior', *IEEE Intelligent Systems* **15**(4), 46–56.
- Byrne, R. W. (1999), 'Imitation without intentionality. Using String Parsing to Copy the Organization of Behaviour', *Animal Cognition* **2**, 63–72.
- Cypher, A., ed. (1993), *Watch What I Do: Programming by Demonstration*, MIT Press.
- Dautenhahn, K. (1994), Trying to Imitate – a Step towards Releasing Robots from Social Isolation, in P. Gaussier & J.-D. Nicoud, eds, 'Proceedings From Perception to Action Conference, Lausanne, Switzerland, September 7-9, 1994', IEEE Computer Society Press, pp. 290–301.
- Dautenhahn, K. & Nehaniv, C. L. (2002a), The Agent-Based Perspective on Imitation, in '(Dautenhahn & Nehaniv 2002b)'.
- Dautenhahn, K. & Nehaniv, C. L., eds (2002b), *Imitation in Animals and Artifacts*, MIT Press.
- Herman, L. M. (2002), Vocal, Social, and Self Imitation by Bottlenosed Dolphins, in '(Dautenhahn & Nehaniv 2002b)'.
- Heyes, C. M. (2002), Transformational and Associative Theories of Imitation, in '(Dautenhahn & Nehaniv 2002b)'.
- Heyes, C. M. & Ray, E. D. (2000), 'What is the Significance of Imitation in Animals?', *Advances in the Study of Behavior* **29**, 215–245.
- Nehaniv, C. L. & Dautenhahn, K. (2000), Of Hummingbirds and Helicopters: An Algebraic Framework for Interdisciplinary Studies of Imitation and Its Applications, in J. Demiris & A. Birk, eds, 'Interdisciplinary Approaches to Robot Learning', Vol. 24, World Scientific Series in Robotics and Intelligent Systems, pp. 136–161.
- Nehaniv, C. L. & Dautenhahn, K. (2001a), 'Like Me? - Measures of Correspondence and Imitation', *Cybernetics & Systems: An International Journal* **32**(1-2), 11–51.
- Nehaniv, C. L. & Dautenhahn, K. (2002), The Correspondence Problem, in '(Dautenhahn & Nehaniv 2002b)'.
- Nehaniv, C. L. & Dautenhahn, K., eds (2001b), Special issue on 'Imitation in Natural and Artificial Systems' of *Cybernetics and Systems: An International Journal*, Vol. 32 (1-2), Taylor & Francis, ISSN: 0196-9722.
- Pepperberg, I. M. (2002), Allospecific Referential Speech Acquisition in Grey Parrots (*Psittacus erithacus*): Evidence for Multiple Levels of Avian Vocal Imitation, in '(Dautenhahn & Nehaniv 2002b)'.