Using Self-Imitation to Direct Learning

Joe Saunders, Chrystopher L. Nehaniv and Kerstin Dautenhahn

Abstract— An evolutionary predecessor to observational imitation may have been self-imitation. Self-imitation is where an agent is able to learn and replicate actions it has experienced through the manipulation of its body by another. This form of imitative learning has the advantage of avoiding some of the complexities encountered in observational learning such as the correspondence problem. We investigate how a system using self-imitation can be constructed with reference to psychological models of motor control including ideomotor theory and ideas from social scaffolding seen in animals to allow us to construct a robotic control system. The system allows a human trainer to teach a robot new skills and modify existing skills. Additionally the system allows the robot to notify the trainer when it is being taught skills it already possesses. We argue that this mechanism may be the first step towards the transformation from self-imitation to observational imitation. We demonstrate the system on a physical Pioneer robot with a 5-DOF arm and pan/tilt camera which is taught using self-imitation to track and point to coloured objects.

I. INTRODUCTION

Social or observational learning is difficult to replicate in artificial systems. Two reasons, among others, for this difficulty is firstly, that to learn from observing the actions of others a mechanism is needed which transforms the others' actions into the same action frame as oneself. Thus a mapping needs to be created between the imitator and the entity being imitated (the model). Secondly, in the making of this mapping it is necessary to know which parts of the body of the imitator are supposed to match those of the model. For humans this matching process can be based on the similar morphology of each person. However, between agents with differing embodiements the matching process is less obvious. Both of these issues are examined in the "Correspondence Problem" [18] and are dealt with in differing ways by different practitioners. However, one simple way to avoid the correspondence problem entirely is to learn from internal observation rather than external observation. This process of internal observation is called *self-imitation* and involves learning from actions made by oneself or made by another on oneself through the process of *putting through*. For example, when teaching children to write the fingers of the child are often physically placed by the teacher around the pencil. This physical process is called *putting through* and allows the child to experience the correct way to hold the pencil, a task that would be difficult to learn from observation alone. In order to use the pencil again the child must replicate the

The authors are members of the Adaptive Systems Research Group, Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, Herts, AL10 9AB, UK. Email: j.2.saunders@herts.ac.uk

motor actions it experienced when being taught and thus self-imitate his or her own physical actions.

In this paper we first report on how self-imitation could be an evolutionary precursor [17] to the more complex stages of imitation and cross-modal imitation. We then review how the social aspects of teaching, learning and self-imitation are used by some social animals to expand their repertoire of skills and define the developmental concepts of *putting through* and *self-imitation* as mechanisms which may prove useful for system teaching. The realisation and validation of our architecture based on these insights and implemented on physical Khepera [1] miniature robots has been described in [25] where we demonstrated how the careful construction of the teaching environment could augment the algorithmic selection of appropriate sensory states experienced in the self-imitation process. Here we describe a further application which has been enhanced to allow the system to notify the trainer of its existing competencies. The system is able to notify the trainer by making predictions of a possible next action based on the set of previously taught behaviours and given the current sensory state. We discuss how this prediction mechanism could prove useful for observational learning given the necessary mappings between imitator and model.

The system is realised on a Pioneer P3-DX [2] robotic platform using a pan/tilt camera and a robotic arm. We firstly teach the robot to track a coloured object using the pan/tilt camera. We then teach the robot to touch the object when the object comes within range of the robotic arm. We describe the scaffolding steps used in building the hierarchy needed to accomplish this and demonstrate how the robot can aid the trainer by indicating when it already knows how to carry out a particular task.

Finally we describe issues that we have recognised during these studies and discuss the possible directions for further research based on these issues and the framework described.

II. EVOLUTIONARY ASPECTS OF SELF-IMITATION

In his proposal on the differing evolutionary paths of imitative learning in primates and psittacine birds, Moore [17] outlines a six-step hypotheses. The process starts with Thorndikian conditioning where existing motor actions are associated and reinforced based on particular environmental conditions. This step is later enhanced by operant (or Skinnerian) conditioning where novel motor responses are formed based on combinations of existing actions. The next evolutionary step is an implicit reinforcement cycle leading to "skills" where the animal is able to perfect the novel

act. The fourth stage introduces the teacher. The teacher essentially guides the pupil by physically *putting through* the actions of the pupil given particular environmental stimuli. This can be considered as *self-imitation* by the animal as it repeats the actions that it has experienced. Visual imitation of others is the next evolutionary stage. In this case the animal now only has to see an act to be able to repeat it. The final process is called cross-modal imitation where an animal is able to match features of its body with corresponding features of another animal. All of the subsequent stages are built and re-inforced on the previous stages. Moore states:

"...these three processes (skill learning, putting through and visual imitation) are linked in many ways: their possible controlling stimuli are nested as just described: both putting through and imitation incorporate and set the stage for skill learning. Putting through is like self-imitation. And all three processes involve novel responses and possibly implicit reinforcement" [17, p.258].

The study presented in this paper bases its mechanisms for system teaching on the *self-imitation* stage. In our current research we do not use any form of implicit reinforcement. Rather, we use explicit reinforcement through teaching of a system by a human. The teaching is carried out using two mechanisms. Firstly the physical act of putting through, where the system's actions in a given environmental state are moulded by the teacher. Secondly, by the process of *scaffolding* where the teacher ensures that the appropriate environmental conditions exist to amplify the learning experience.

III. Self-Imitation, Putting Through and Scaffolding in Animals

Evidence for self-imitation from teaching using putting through and scaffolding in the animal kingdom come mainly from studies of primates [8]. However there is also evidence from carnivores including domestic cats, tigers, cheetahs, otters, dolphins, orca whales and some bird species [26].

For example Fouts *et al.* report on the chimpanzees Washoe and Loulis, Loulis being the adopted infant chimp of the mother Washoe. Washoe had previously been taught American Sign Language (ASL) however the human carers made no attempt to teach Loulis ASL and did not use ASL in Loulis' presence. However Washoe succeeded in teaching Loulis ASL both by demonstration and by *putting through* of Loulis' hands [22]. This technique had also been used by the human carers to originally teach Washoe. Animals have also been observed modifying the environmental conditions experienced by a learner. This process, called *scaffolding*, is used typically by the mother, to make it much easier for her child to complete the task when the child is at a developmental stage where it could not perform the appropriate acts or sequence its actions correctly. Scaffolding of tasks together with observational learning and putting through have been observed in wild chimpanzees [8]. For example, cracking nuts with a hammerstone is an especially difficult task for a chimpanzee to learn, taking up to 14 years to perfect in some cases. A number of observations have been recorded where

the mother will clean the anvil, reposition the nut or re-orient the hammerstone to favourable orientations for the infant. Additionally mothers will often leave their hammers and a supply of nuts in favourable positions for their young to use when normally adult chimpanzees would eat the available nuts and retain their hammers.

Scaffolding is also a familiar concept in human development and is emphasised in Vygotysky's idea of the "zone of proximal development" in his theory of the child in society [28]. Vygotsky emphasised the idea that teaching and social interaction allow higher competence levels to be achieved through staged learning and building upon existing skills. Vygotsky (and also Piaget) both argue that the learner learns based on their own sensorimotor experiences, their own activity is at the centre of the learning process.

We take inspiration from these examples in social animals to study how self imitation via *putting through* and *scaffolding* can be used to good effect in teaching robots to learn new skills and modify existing skills.

IV. Imitation Perspective

Approaches to imitative learning can be classified into specialist and generalist theories [7]. The generalist theories, such as ideomotor theory (IM) [20] and associative sequence learning (ASL) [13], assume that imitation is mediated by learning and motor control. This is in contrast to the specialist theories, such as active intermodal matching (AIM) [15], which proposes special purpose mechanisms for imitation. Our approach is closest to that of extended ideomotor theory [20] where the defining feature of the imitation attempt is the idea that the similarity between an event perceived by the imitator and an event learned from the imitator's own actions will induce that action. This idea of similarity is central to imitation and we use it in the learning algorithms implemented on our systems. Previous learning is what gives rise to matching actions and we match perceived internal and external perceptions with motor actions in a single form of memory representation.

V. Approach and Related Work

As noted above imitative learning faces two difficult problems. Firstly, that by observation alone the proprioceptive feedback that the teacher experiences cannot be directly experienced by the pupil [24] and secondly, there may be a mismatch between the external and internal sensorimotor spaces of the teacher and pupil - the correspondence problem [18]. Our approach is to use supervised learning to learn a control policy directly. The correspondence issue is avoided by having the pupil experience the same set of actions and sensory states as the teacher simply by the process of putting through. Thus there is no need for observational matching and similarly there is no correspondence problem as the system is corresponding directly with itself. The system learns to associate the actions moulded on its body against perceived environmental state. Currently our systems have no mechanism for proprioceptive feedback, therefore we

Fig. 1. The environment showing the Pioneer robot with pan/tilt camera and 5-DOF arm.

control the movements of the system directly using tele-operation. This form of tele-operation being a proxy for the more direct form of putting through where the human would directly manipulate the system's actuators. During the process of putting through the human trainer has no access to the system's internal state or perceptions and in fact may not be aware of the form of sensorimotor feedback that the system is experiencing.

For learning we use an approach described by Bentivenga *et al.* [5] where a framework is constructed using a memory based *k*-Nearest Neighbour machine learning technique. This technique is also related to loose-perceptual matching methods described in [3]. The concept behind using memory based approaches is that intelligent behaviour is obtainable from analogical reasoning i.e. that we can extrapolate behaviour to new situations based on the similarity of stored representations of previous behaviour. A fundamental advantage of this approach is that errors made during the training process can be corrected simply by providing further additional and correct training experiences. This is in contrast to behavioural cloning [23], [27] and production rule approaches [11] to learning by demonstration where both good and bad training experiences are stored with equal weight and are not easily corrected. Early work using the ideas of direct manipulation via *putting through* can be found in [4] where hybrid control programs are generated from teaching data supplied from a human manipulating a robotic end-effector.

Our approach is detailed in [25] and parts are repeated here for clarity. We extend this work by allowing the system to *predict forthcoming actions and inform the trainer* if it already knows how to carry out a particular task.

VI. FRAMEWORK

For this study our experiments are based on training a system to point to coloured objects. The system comprises a Canon VCC4 Video camera with a pan/tilt unit mounted on a Pioneer P3-DX robotic platform [2] with a 5-DOF arm. The sensory feedback to the learning mechanism is the centre point and area or the region of interest of a chosen object togther with the current modality of the pan/tilt unit and the robotic arm. We pre-train the system to detect

coloured objects before applying our framework. The object is detected and tracked in the camera frame using the CAMSHIFT [6] algorithm, part of the Intel OpenCV image processing library [19].

In the following sections we describe:

A. how the robot uses similarity measures to learn.

B. how *putting through* is used in matching robot states and trainer directed actions.

C. how relevant attributes in the robot state are selected and amplified by *scaffolding*.

D. how the robot is able to learn new tasks and modify existing tasks.

E. how the robot uses a mechanism of action prediction to inform the trainer when it has an existing competence.

A. Learning Mechanism

We use a memory based "lazy" learning method [16] to allow the system to learn tasks. This is a *k*-nearest neighbour (kNN) approach where the value of each feature in the system's state vector (see *Scaffolding* below) is regarded as a point in *n*-dimensional space, where *n* is the number of features in the state vector (see table 1). For each chosen task we collect a set of training examples (as described in *Putting Through* below) together with their target primitives, each primitive being chosen by the human trainer when moulding the system's actions. We call this collection of states a *memory model*. When the task is executed the system continually computes its current state vector. It then computes the distance from the current state to each of the training examples held in the memory model. The distance between the state vector and the training example being the sum of the distances between the features in each, as follows:

$$distance(X, S) = \sum_{i=1}^{n} W_i \left| \frac{x_i - s_i}{max_i - min_i} \right|$$

Where *X* is an instance of the training examples and *S* an instance of the system's current sensory state. *W* is a non-negative vector of real numbers used to weight each of the dimensions. This weighting is discussed in the *scaffolding* section below. Setting *k* to 1 will result in the nearest point in the training examples being used and yield a single primitive as its target function. Where *k* is greater than 1 the algorithm will yield a set of primitives. We choose the most common primitive from the set as the target function. Note that this method will always result in a primitive being chosen. In environmental situations not previously experienced by the system, generalisation occurs as the primitive nearest to the current state is chosen. Thus performance is based on the similarity of new situations to those already experienced.

In work to date the *k* value has been set experimentally to approximately correlate to the number of state vector entries in each memory table. In forthcoming work we intend to compute the *k* value using cross-validation. Currently, for a small number of state vector entries *k* is set to 1. For larger numbers of entries *k* has been set to higher values but to date not exceeding 5. We make use of the Tilburg University Memory Based Learner [9] to provide the kNN functionality.

TABLE I

State	Description
Pan	Current Pan Setting
Tilt	Current Tilt Setting
X-value	Location of object in camera X-direction
Y-value	Location of object in camera Y-direction
Delta-X	Change in object X-direction from last frame
Delta-Y	Change in object Y-direction from last frame
Area	Size of object tracking area
Sonar	Values of the 5 front and 5 rear sonars
Bumpers	Values of the 16 bump sensors
Arms	Angles of 6 arm joints

TABLE II

Pre-defined Primitives.

Primitive	Description
Pan Left	Pan Left 5° or continuously
Pan Right	Pan Right 5° or continuously
Tilt Up	Tilt Up 5° or continuously
Tilt Down	Tilt Down 5° or continuously
Move Forwards*	Move Forward 10cm or continuously
Move Backwards*	Move Backwards 10cm or continuously
Turn Right*	Turn Right by 5° or continuously
Turn Left*	Left Left by 5° or continuously
Increase Joint Angle	Increase a given joint angle by 5°
Decrease Joint Angle	Decrease a given joint angle by 5°
*	(=Not used in these experiments)

This system has the advantage of providing a very efficient kd-tree based coding structure for the training examples so as to speed up performance.

B. Putting Through

The concepts of scaffolding and putting through can play an important part in animal learning. They support a form of self-imitation that may be the natural precursor to more complex forms of imitative learning. In our framework we use the idea of putting through directly. The human has the ability to control the system by remotely moving it through a set of pre-defined basic primitives. This set of primitives are basic actions available to the system (see table 2). The human teacher has no access to the internal state of the system. By manipulating the system in this manner we also avoid both the problem of observation by the system of the human actions and of the correspondence problem between the system and human.

During the system moulding process a snapshot of the systems proprioceptive and exterioceptive state (see table 1) is recorded together with the directed primitive on each human command to the system. For each human defined task we can therefore build a memory model of state/primitive combinations.

C. Scaffolding

All of the states perceived by the system are recorded in the state vector however particular attributes may have more relevance to different tasks. For example, in this experiment the values of the X-Y object tracking values are of more relevance than the values of the bumpers (which remain constant).

We use two mechanisms to ensure that the appropriate attributes are chosen. The first is based on computing *information gain* to measure how well a given attribute separates the set of recorded state vectors according to the target primitive. This is defined as follows:

$$Gain(S,A) = Entropy(S) - \sum_{v \varepsilon Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where *S* is the collection of training examples, *Entropy(x)* is a function returning the entropy of *x* in bits, *Values(A)* is the set of all possible values for a particular state attribute *A* and S_v is the subset of *S* for which attribute *A* has value *v*. Further explanations of this metric can be found in [21], [16]. The information gain measurement allows particular attributes in the state vector to have greater relevance by using it to weight the appropriate dimensional axes in the kNN algorithm (by setting W_i above). This has the effect of either lengthening or shortening the axes in Euclidean space thus reducing the impact of irrelevant state attributes.

The second mechanism for attribute selection is the human trainer. It is assumed that the trainer already understands the task (from an external viewpoint) that the system must carry out and therefore is able to construct the training environment appropriately so as to ensure that irrelevant features are removed. This idea allows the technical selection of relevant state features to be enhanced as the other features will now tend to have constant values and therefore a low information gain. As discussed in section 3 above this process of scaffolding or creating favourable conditions for learning would seem a quite natural phenomenon in social animals and is of course fundamental to all forms of human teaching.

D. Learning New Tasks

The trainer directs the system (in this case the camera pan/tilt unit and robotic arm) using a screen based interface which provides a number of buttons used to set operation modes such as "execute" and "start/stop learning" plus an edit field to label actions and a list from which to choose existing labelled actions and primitive operations.

The system can be in two modes. The first is *execution mode*, which is its normal mode of operation where its current behaviour is executed. Alternatively the system can be in *learning mode* where the human trainer can put through, scaffold and create new activities for the system to eventually use in execution mode.

In "learning" mode the system can be taught new competences: *sequences*, *tasks* or *behaviours*. All three learning levels are started by pressing a "start learning" button and terminated by pressing a "stop learning" button. For each new competence (either a behaviour, task or sequence) the trainer explicitly provides an appropriate label. When training is complete the label is added to the set of actions available to the trainer and thus can be used immediately for further training sessions.

Existing labelled actions can also be modified (or entirely deleted) with additional training episodes as required.

The *sequence* level is where the system can be directed through a given sequence of primitives which it records without reference to its state i.e. sequences are entirely independent of the internal or external environment. An example of a sequence might be to move the arm to a particular position. This could, for example, be labelled as the 'readyArm' sequence. The readyArm sequence would then become part of the available set of competences available for the trainer to use. These new sequences could then be used in combination with other primitives and other sequences to create further sequences.

The *goal-directed task* level differs from a sequence in that during training the actions taken by the system will depend on the the robot's internal and external state at that time. The trainer now has the opportunity to select not only basic primitives, but sequences and other goal-directed tasks. The tasks are goal-directed because the trainer is able to inform the system when the task has completed with the resulting state being recorded as a target to achieve. This goal condition is paired with the system state and becomes a further training record in the memory model for that particular task. As an example consider a tracking behaviour, for which the goal state is to have the target centred in the visual field. The trainer could pan the camera to the right, choose the "goal-directed task" level, label it "pRight" and press the "start learning" button. The system can then be put through a right panning situation tracking an object. The trainer would signal that the goal state was reached when the camera lens was directly in front of him/her. This training regime would be repeated for many panning situations and thus many panning recognition states with appropriate actions and goal states being recorded into the 'pRight' memory model. Note that the signalling by the trainer to the robot of a goal-state does not automatically imply that teaching has stopped, it simply signals that this state is a goal-state. The trainer could continue training the robot by placing it in situations that are not goal-states. The "stop learning" button is only pressed once the trainer is happy with the training regime. Similarly an existing training episode can be enhanced with further training episodes by choosing an existing task and pressing the "start learning" button.

The *behaviour* level allows the trainer to construct the complete behaviour for the system from the component set of tasks, sequences and primitives. The construction of a behaviour is the same as for a task except that no goal state is required. The behaviour will run continually in execute mode and base its decision of what task, sub-task, sequence or primitive to use based on the current environmental state. With careful training the trainer can now build a hierarchy of tasks, sequences and primitives as required (see figure 2).

E. Task Prediction

Both the goal directed tasks and the overall behaviour depends on the robot state. When the teacher is training the robot each action is associated with the scaffolded state at that time and for each trained component a memory model
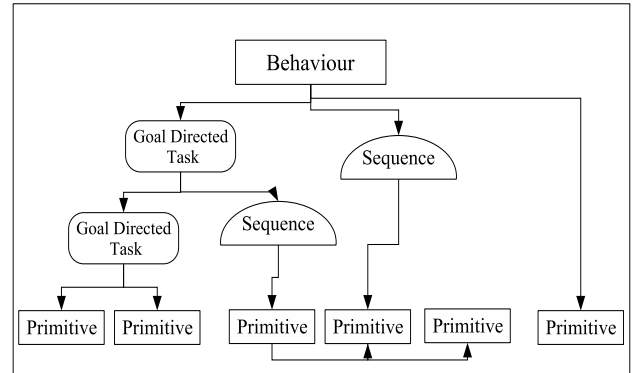

Fig. 2. An example of a trained hierarchy of primitives, primitive sequences, learned goal-directed tasks and behaviour.
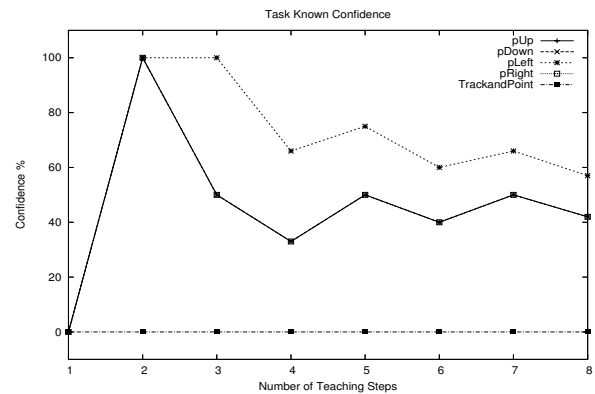

Fig. 3. Recognition of previously learned behaviour using Task Prediction. The graph shows how well each task or behaviour matches the current training sequence. As each training step is taken the task/behaviour most similar to that being trained increases in confidence. The first training step achieves 100% confidence as each task in this step matches the goal condition. However as more training episodes are added the pLeft task maintains a higher confidence level as predicted actions match trained actions. The peaks in the confidence level is due to the trainer using a slightly different training regime.

is created. As well as being used to control overall behaviour these memory models can be employed to predict whether the trainer is teaching tasks already known to the robot. Based on the current state each memory model is polled using the kNN algorithm above. This polling will yield a set of possible actions, one for each memory model in the system. If the trainer directs (puts through) the robot to take one of these actions, a weight attached to each memory model that proposed this action is incremented. As each training step is taken this cycle is repeated. Once the weight of any of the memory models exceeds a threshold the trainer is informed by the robot that it may already have knowledge of this task. An example of this process is shown in figure 3. This process is in some ways similar to the prediction capabilities of imitation systems using forward models [10], [14]. In these systems a set of forward models predict a set of possible forthcoming states which are weighted by matching against current observed (and transformed) states. This eventually leads to the appropriate inverse model (which is paired with a forward model) being selected as the

imitative action. In this study we instead match predicted actions against actions put through by the trainer and weight the single memory representation for each task accordingly. The process differs in that the observation is internal rather than external, that actions rather than states are matched and that each task model is represented in one structure in line with the ideas of ideomotor theory outlined above.

VII. EXPERIMENTAL VALIDATION

We illustrate the successful functioning of the architecture in a tracking and pointing task. The trainer is required to teach the robot to track a coloured object held in the trainer's hand using the camera pan/tilt unit. If the object comes within range of the extent of the robot arm, the arm is trained to move and point to the object. When out of range the arm is held in a "ready" or "home" position. Prior to training the robot has no abilities other than the pre-defined primitives shown in table 2. The final training hierarchy that results from the training exercise is shown in figure 4. Note that for reasons of clarity figure 4 shows only each unique sequence, task or primitive per memory model. In reality each memory model may have a great many instances of different states in each sequence, task or primitive.

The first step in the teaching process is to amplify the selection of appropriate sensorimotor experiences by correctly scaffolding the environment. In this case this means ensuring that extraneous effects which might affect the sensorimotor space are avoided by placing the robot in a space unaffected by other robots or people. We then construct the learning experience in a series of stages. Firstly a series of enhanced primitives are created using the sequencing system. For example, we want to keep the arm in the same orientation as the camera pan-tilt unit. To achieve this a new primitive is created called ArmPanTiltRight which comprises of a right arm movement on the first joint angle and a right move of the pan/tilt unit. We create a similar ArmPanTiltLeft. Note that as the actuators are independent the primitive commands, although issued sequentially, are activated by the motors in parallel. Secondly, for each direction (up,down,left,right) we train a separate goal-oriented task. This separation of directions is to further scaffold the training experience because, if combined, the selection of appropriate attributes (the pixel X-Y location of the object) would be difficult as the X attribute is important for left-right but it is the Y attribute that is important for up-down (also see the *Discussion* section below). We use the newly created sequences to do this and create tasks such as pLeft or pRight. Thirdly, we create two sequences which move the arm to a pointing position and one which moves the arm to a resting position. These sequences are activated by the high level behaviour by training it to execute the pointing primitive if the object is held close to the camera but otherwise use the resting arm movement. The high level behaviour is also trained to execute the panning tasks when the object is in each quarter portion of the XY plane.

During the the training process the prediction mechanism is always on. Thus if the trainer were to attempt to train
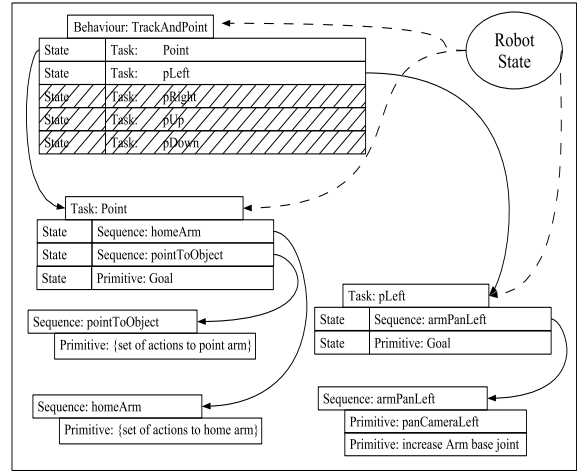

Fig. 4. The hierarchy built through the teaching process. The top level behaviour uses the robots sensorimotor state to choose the most similar state and execute the appropriate trained function. This process is iterated down the hierarchy until a primitive is executed. For clarity only the first two levels of the behaviour are expanded, the hatched area has not been expanded.

the robot to pan left in response to the object (i.e. create a new version of pLeft), the system would respond with a high similarity measure as further actions are introduced. This is shown in figure 3 where a new version of pLeft is trained. Initially four behaviours achieve 100% confidence as they all match the goal condition. However as more training steps are taken only the pLeft task remains at a higher confidence level. The robot informs the trainer of a high match via messages sent to a display box on the control system.

Following training, which is carried out in an iterative manner, both correcting errors and testing tasks throughout, the robot successfully tracks and points to particular coloured objects when presented to it by the trainer.

VIII. DISCUSSION

We have demonstrated how a robotic social learning system can be constructed which uses the concepts of self-imitation and task and environmental scaffolding. The system has been implemented and successfully demonstrated on two different physical robot architectures, the first described in [25] and second discussed here. The latter architecture has been extended with a predictive capability which aids the teacher in understanding the robot's current capabilities.

During our experiments we have recognised issues that require further research. Firstly, the system has no mechanism for recording historical states i.e. has no form of long or short term memory. This limits its use in that previous events that the teacher may wish to use to inform current actions is not possible as they are not part of the similarity state space of the robot. Secondly, although we have made efforts to use unmodified sensory information wherever possible there are instances when this form of sensory capture would overwhelm and may be inappropriate for the learning model, for example, the capture of unadulterated camera images. We therefore use some prior knowledge of the expected

applications to modify the sensory stream when necessary. Thirdly, we are limited in our hardware in having to use tele-operation as a proxy for direct manipulation of the robot due to the lack of proprioceptive feedback from the robot's actuators. This means that training steps tends to be discrete and small in number as the interface cannot replicate continuous movements. The result is that the level of detail available to us in this system is 'coarse' rather than 'fine' and as such much useful information may be missing. Finally, in the current model each sensor stream is recorded as a unique attribute in the kNN classification algorithm. However, when weighting the attributes using the information gain criteria, relevant attributes may be underweighted as they are only partly relevant to the classification at a given time. These and other issues form part of our ongoing research.

A feature of the system is that the robot predicts which task is most similar to the one that the trainer is currently teaching by matching proprioceptive actions against predicted actions given the current sensorimotor states. This predictive facility based on similarity may be a precursor to observational imitation if the idea is extended to associate learned state and action pairs against actual effects (although the thorny problem of correspondence remains). This extension would allow for a predictive similarity measure between perception of an event and the learned action-event pair in a single form of memory representation. Thus there is no distinction between either perception and action, and therefore perception can be used as the similarity measure leading to motor actions. As Prinz puts it "to select a certain act, given an intention to achieve certain effect" [20, p.143]. The inverse is the similarity of motor actions informing perception and therefore "one is to expect certain effects given certain acts" [20, p.143]. This idea follows directly from the extended definition of ideomotor theory and is also closely related to the perspectives used in control engineering for forward and inverse models [12], [10]. We intend to pursue this extension in further research using this architectural model.

IX. ACKNOWLEDGMENTS

REFERENCES

[1] K-Team Corporation, Y-Parc - rue Galile 9, Yverdon-les-bains 1400, Switzerland.

[2] Mobile Robots Incorporated,19 Columbia Drive, Amherst, NH 03031, USA.

[3] A. Alissandrakis, C. L. Nehaniv, K. Dautenhahn, and J. Saunders. An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated objects. In *6th IEEE Int. Symp. Computational Intelligence in Robotics and Automation (CIRA'05)*, pages 61–66. IEEE, 2005.

[4] H. Asada and H. Izumi. Automatic program generation from teaching data for the hybrid control of robots. *IEEE Transactions on Robotics and Autonomation*, 5(2):166–173, 1989.

[5] D. C. Bentivegna and C. G. Atkeson. A framework for learning from observation using primitives. In *Proc. RoboCup Int. Symp., Japan*, 2002.

[6] G. Bradski. Computer vision face tracking as a component of a perceptual user interface. In *Workshop on Applications of Computer Vision*, page 214219, Princeton, NJ, October 1998.

[7] M. Brass and C. M. Heyes. Imitation: is cognitive neuroscience solving the correspondence problem? *Trends in Cognitive Science*, 9:489–485, 2005.

[8] R. W. Byrne. *The Thinking Ape: Evolutionary Origins of Intelligence*. Oxford University Press, 1995.

[9] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. Timbl:tilburg memory-based learner. Technical Report ILK 04-02, Tilburg University, 2004. Available from http://ilk.uvt.nl/.

[10] 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. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 327–361. MIT Press, 2002.

[11] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47:109–116, 2004.

[12] M. Haruno, D. M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220, 2001.

[13] C. M. Heyes. Transformational and associative theories of imitation. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 501–523. MIT Press, 2002.

[14] M. Johnson and Y. Demiris. Hierarchies of coupled inverse and forward models for abstraction in robot action planning, recognition and imitation. In *Proceedings of 3rd International Symposium on Animals and Artifacts at AISB 2005*, pages 69–76. SSAISB, The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, April 2005.

[15] A. N. Meltzoff. Imitation as a mechanism of social cognition: Origins of empathy, theory of mind, and the representation of action. In U. Goswami, editor, *Handbook of Childhood Cognitive Development*. Blackwell Publishers, 2002.

[16] T. M. Mitchell. *Machine Learning*. McGraw-Hill International, 1997.

[17] B. R. Moore. *Social Learning in Animals: The Roots of Culture*, chapter The Evolution of Imitative Learning, pages 245–265. Academic Press Inc., 1996.

[18] C. L. Nehaniv and K. Dautenhahn. The Correspondence Problem. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 41–61. MIT Press, 2002.

[19] OpenCV. http://www.intel.com/technology/opencv/index.htm. Last referenced on 16th of March, 2006.

[20] W. Prinz. An ideomotor approach to imitation. In S. Hurley and N. Chater, editors, *Perspectives on Imitation*, volume 1, pages 141–156. MIT Press, 2005.

[21] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[22] R.S.Fouts, D.H.Fouts, and T. Cantfort. *Teaching sign language to chimpanzees*, chapter The infant Loulis learns signs from cross fostered chimpanzees, pages 280–92. State University of New York Press, 1989.

[23] C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In *Proc. Ninth Int. Conf. on Machine Learning*, pages 385–393. Morgan Kaufmann, 1992.

[24] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. An experimental comparison of imitation paradigms used in social robotics. In *Proc. IEEE Robot and Human Interactive Communication (ROMAN '04)*, pages 691–696. IEEE Press, September 2004.

[25] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment. In *Proceedings of the ACM Conference on Human-Robot Interaction (HRI06)*, pages 118–125. ACM Press, March 2006.

[26] T.M.Caro and M.D.Hauser. Is there teaching in non-human animals? *Quarterly Review of Biology*, 67:151–74, 1992.

[27] M. van Lent and J. E. Laird. Learning procedural knowledge through observation. In *K-CAP 2001: Proc. Int. Conf. Knowledge Capture*, 2001.

[28] J. V. Wertsch. *Vygotsky and the Social Formation of the Mind*. Harvard University Press, 1985.