ARS ADVANCED ROBOTIC SYSTEMS INTERNATIONAL

Self-imitation and Environmental Scaffolding for Robot Teaching

Joe Saunders, Chrystopher L. Nehaniv, Kerstin Dautenhahn and Aris Alissandrakis
Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire,
Hatfield, Herts., AL10 9AB, United Kingdom
Corresponding author: Joe Saunders, J.2.Saunders@herts.ac.uk

Abstract: *Imitative learning and learning by observation are social mechanisms that allow a robot to acquire knowledge from a human or another robot. However to be able to obtain skills in this way the robot faces many complex issues, one of which is that of finding solutions to the correspondence problem. Evolutionary predecessors to observational imitation may have been self-imitation where an agent avoids the complexities of the correspondence problem by learning and replicating actions it has experienced through the manipulation of its body. We investigate how a robotic control and teaching system using self-imitation can be constructed with reference to psychological models of motor control and ideas from social scaffolding seen in animals. Within these scaffolded environments sets of competencies can be built by constructing hierarchical state/action memory maps of the robot's interaction within that environment. The scaffolding process provides a mechanism to enable learning to be scaled up. The resulting system allows a human trainer to teach a robot new skills and modify skills that the robot may possess. Additionally the system allows the robot to notify the trainer when it is being taught skills it already has in its repertoire and to direct and focus its attention and sensor resources to relevant parts of the skill being executed. We argue that these mechanisms may be a first step towards the transformation from self-imitation to observational imitation. The system is validated on a physical pioneer robot that is taught using self-imitation to track, follow and point to a patterned object.*
Keywords: *Social Robotics, Imitation, Teaching, Memory-based learning, Scaffolding*

1. Introduction

What differentiates a robot from a collection of specialised machines? Why would a consumer choose to purchase a domestic robot rather than a smarter dishwasher or an automatic laundering system? One answer may be that the robot might be clever enough to carry out a *range of tasks* and adaptable enough for the purchaser to *train* it to carry out new tasks. For example, imagine the robot is delivered with pre-programmed behaviours which allow it to carry out useful tasks around the home e.g. collecting cutlery, cups and plates and placing them in a dishwasher or tidying up by picking up clothes left on the floor and placing them in a washing basket. However, although the robot performs to the manufacturer's specifications there are some tasks which it does not carry out. It fails to tidy up the children's toys into the toy cupboard or it fails to recognise that a particular and expensive glass should not be placed in the dishwasher. Rather than call in the manufacturer to re-program the robot the purchaser may prefer to show the robot what to do, effectively teaching the robot to carry out new skills or to modify existing skills. A simple and intuitive method that would allow humans to train and shape robot behaviour is clearly a primary goal in making this task easier. In researching how to meet this goal two issues arise; firstly how should the robot learn these new skills or modify its existing skills and secondly, how should the trainer teach the robot new or modified skills.

Ideally we might wish the robot to observe what we do and imitate it. However 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 made 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 embodiments the matching process is less obvious. This issue is referred to as the "Correspondence Problem" (Nehaniv and Dautenhahn 2002). Further difficulties are that in some cases learning through observation alone will be insufficient to correctly learn a skill. For example, teaching a robot to pick up a wet glass will be different from teaching it to pick up a dry glass. The robot may only learn the correct procedure from having directly experienced the skill as there may be insufficient information available from observation alone (Saunders, Nehaniv and Dautenhahn 2004). It is also possible that a task may not involve a straightforward matching of movements. The task may be sufficiently complex that the robot will need to be shown each sub-task separately and then shown how to combine these

into more complex behaviours. This form of teaching and skill assembly reflects many of our own human experiences when being taught new skills.

In this paper we examine these issues and propose, implement and validate an architecture that allows a robot to learn, via supervised teaching, new tasks and be able to modify existing tasks. The system proposed places emphasis on how a robot can learn, how it can be helpful by providing feedback to the trainer during learning and how it can exploit what it has learned to optimise its sensory and attentional mechanisms and thus operate more efficiently. We also highlight the role of the teacher in this process, whereby the teacher can carefully structure the learning experience and enable the robot to learn more effectively.

We start this discussion by considering how to avoid the correspondence problem by using a technique of learning from internal observation rather than external observation. Matching behaviour based on this process of internal observation is called *self-imitation* and involves learning from actions made by oneself or made by another on oneself. 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 motor actions it experienced when being taught and thus self-imitate his or her own physical actions. We review how this idea of *self-imitation* may have been an evolutionary precursor (Moore 1996) to the more complex stages of imitation and cross-modal imitation.

We then consider 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 concept of *scaffolding* as a mechanism which may prove useful for scaling up complexity in robot teaching.

The realisation and validation of our architecture based on these insights has been implemented on two physical[1] robotic platforms. The first, using Khepera miniature robots (k-Team 2006), demonstrated how the careful construction of a teaching environment can augment the algorithmic selection of appropriate sensory states experienced for learning in the self-imitation process (Saunders, Nehaniv and Dautenhahn 2006). In the second, which is described here, an enhanced version of this application is realised on a "human-scale" Pioneer P3-DX robotic platform (MobileRobots 2006). This platform has a larger sensory-motor space than the Khepera and illustrates how the methods can scale to more complex, real world robotic systems. This realisation of the learning architecture also allows the robot to notify the trainer of its existing competencies and to ensure that its focus of sensory attention is optimised for the taught task. The robot 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 mechanism could prove useful for observational learning given the necessary mappings between imitator and model. The architecture is validated using the Pioneer on a task which involves the robot perceiving, camera tracking and then physically following a patterned object. Working versions of the architecture on the Pioneer have been successfully demonstrated and trained at two live events, (see Saunders (2006) for details).

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

2. Evolutionary Aspects of Self-Imitation

In his proposal on the evolutionary path of imitative learning, Moore (1996) describes six key steps. The learning 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. For example, human babies touch parts of the faces of their parents and can then locate the same features on their own face. All of the subsequent stages are built and reinforced 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

[1]*Throughout this research we use physical, situated and embodied robots. We do not use software simulations or any form of external sensory feedback (such as external cameras) to the robots.*

responses and possibly implicit reinforcement" (Moore 1996 [p.258]).
The study presented in this paper bases its mechanisms for robot 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. Our previous work (Saunders, Nehaniv and Dautenhahn 2004) considered aspects of observational/imitative learning at the visual *imitation* (fifth) stage. Our current work develops these ideas in a further investigation of the spectrum of possible learner/model social learning interactions.

3. 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 (Byrne 1995). However there is also evidence from carnivores including domestic cats, tigers, cheetahs, otters, dolphins, orca whales and some bird species (Caro and Hauser 1992).

Fouts *et al.* provide compelling reports 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 (Fouts, Fouts and Van Cantfort 1989). This technique had also been used by the human carers to originally teach Washoe.

Herman (2002) reports on vocal, social and *self-imitation* in extensive studies with bottlenosed dolphins and argues that it is not the origin of the imitative representation, either internal or external, that is relevant but rather that an animal can create a mental representation of a behaviour and replicate that behaviour.

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 have been observed in wild chimpanzees (Byrne 1995). 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.

Another example of scaffolding from cheetahs is where the mother rather than killing prey will capture and release the live prey to the cheetah cubs when they are about 3 months old. The behaviour is also selective, only prey species which the cubs are likely to catch are released. It appears that the cubs' experience results in faster learning and a more skilled performance. This was tested with domestic cats whose kittens were brought live mice by their mothers at an early age. By 6 months old the kittens showed superior skills to a test group who had not been exposed to the mice (Caro 1980).

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 (Wertsch 1985). Vygotsky emphasised the idea that teaching and social interaction allow higher competence levels to be achieved through staged learning and building upon existing skills. This contrasts with the Piagian view of the child as a solitary thinker or little scientist exploring the world (Piaget 1945/1962). However Vygotsky and Piaget both argued that the learner learns based on their own sensorimotor experiences; their own activity is at the centre of the learning process.

The examples above illustrate that demonstration, putting through and scaffolding can be used when there is a specific lack of knowledge in the pupil. The teacher is aware that the task to be learned may not be directly acquired but requires some developmental building blocks, these blocks allowing the pupil to reach a further developmental stage.

We take inspiration from these examples in humans and other 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.

4. Imitation Perspective

Approaches to imitative learning can be classified into specialist and generalist theories (Brass and Heyes 2005). The generalist theories, such as ideomotor theory (IM) (Prinz 2005) and associative sequence learning (ASL) (Heyes 2002), assume that imitation is mediated by learning and motor control. This is in contrast to the

specialist theories, such as active intermodal matching (AIM) (Meltzoff 2002), which proposes special purpose mechanisms for imitation. Our approach is closest to that of extended ideomotor theory 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 (Wohlschlager, Gattis and Bekkering 2003). 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.

5. Approach and Related Work

Even with explicit programming robot control is hard due to sensor noise, the non-deterministic state of the environment, the inability to ensure that the robot's actions are deterministic and the need for real-time responses. There are generally a number of problems which need to be solved:

- *How can the human teach the robot?* - what mechanisms can be used to make the robot meet the needs of the teacher? How can the robot learn when the task is complete?
- *What techniques can the robot use to learn?* - how can the robot generalise and execute the new task?
- *How can the robot incorporate the new experiences into its existing competencies?* - what sort of structure is necessary to ensure that competencies for acquiring new tasks can co-exist with competencies at existing tasks?
- *How can it select the right action at the right time?* - given a learned set of competencies which one is appropriate?

Approaches include topics such as *programming by demonstration*, *imitation learning*, *learning from observation* and *robot shaping*.

5.1 Related Work

Typically the observational and imitative approaches attempt to match the behaviour of the demonstrator and so construct an appropriate control policy. Schaal *et al.* (2003) provide a useful overview where approaches to the problem are classified as follows:

i. *direct policy learning* - where supervised learning is used to learn a control policy directly.
ii. *learning policies from demonstrated trajectories* – this assumes that the task goal is known and uses sample trajectories to learn a control policy
iii. *model based policy learning* - where a predictive model of the control problem is constructed.

All of these approaches face two difficult problems. Firstly, that by observation alone the internal proprioceptive feedback that the teacher experiences cannot be directly experienced by the pupil (Saunders, Nehaniv and Dautenhahn 2004) and secondly, there may be a mismatch between the external and internal sensorimotor spaces of the teacher and pupil - the correspondence problem (Nehaniv and Dautenhahn 2002).

In the *direct policy approach* these issues can be avoided by having the pupil experience the same set of actions and sensory states as the teacher with the correspondence problem solved by ensuring that both teacher and pupil have a similar embodiment. This approach has been used by a number of groups, including Billard and Dautenhahn (1999) and Hayes and Demiris (1994). In both cases a student robot followed a teacher robot and learned to associate imitated actions against perceived environmental state. Saunders *et al.* (2005) however have demonstrated that there can be limitations in this approach due to reactive impersistence and teacher interference when using a pure following approach due to the fact that the imitator must stay in close proximity to the teacher in order to learn. However the imitator's ability to directly experience similar sensory cues as the model is a positive feature of this approach.

In work by Nicolescu and Matarić (2001) a mobile robot tracks a teacher's movements by following the teacher and matching predicted postconditions against the robot's current proprioceptive state. It then builds a hierarchical behaviour-based network based on "Strips" style production rules (Fikes and Nilsson 1971). This work attempts to provide a natural interface between robot and a teacher (who provides feedback cues) whilst automatically constructing an appropriate action-selection framework for the robot. Although the framework has been demonstrated to work effectively, the validation tasks chosen have tended to be within a limited experimental set and it is not clear if the framework could be extended to scale and cope with additional and different task sets.

Another way to allow a robot to experience the appropriate sensory-motor states is by allowing the teacher to manipulate the robot directly via a form of tele-operation and record the sensory states of the robot. Although not using a robot, this method is closely related to Sammut *et al's.* (1992) "learning-to-fly" application (generally called *behavioural cloning)* where recordings of control parameters in a flight simulator flown by a number of human subjects were analysed using Quinlan's

C4.5 induction[2] algorithm (Quinlan 1993). The algorithm extracted a set of "if-then" control rules. Van Lent and Laird also used this approach but provided a user interface which could be marked with goal transition information (van Lent and Laird 2001). This allowed an action selection architecture to be constructed using "Strips" (Fikes and Nilsson 1971) style production rules. However, in both of these research areas the full "state" of the system (both internal and external) is available to the trainer. This may not be the case when teaching robots. The approach also tends to result in specialised learning although some researchers (Arentz 1994) have investigated how to introduce noise into the training environment to cope with this.

Early work using the ideas of direct manipulation via *putting through* was carried out by Asada *et al.* (1989) where hybrid control programs are generated from teaching data supplied from a human manipulating a robotic end-effector. Similarly, work by Levas and Selfridge (1984) use a form of tele-operation to construct a set of production rules. In our work tele-operation is used primarily as a proxy for direct robot manipulation rather than the prime mechanism. Also we avoid the use of symbolic reasoning and planning systems, relying instead on the direct sensorimotor experiences of the robot.

A long line of research into teaching service robots by observing humans has also been carried out by Dillman *et al.* (Kaiser, Klingspor, Millán, Accame, Wallner and Dillmann 1995; Kaiser and Dillman 1996; Dillmann 2004) where after observation, production rules are generated to produce grammatical formalisms held in a knowledge database of actions. However, with these and the other symbolic approaches mentioned above it is unclear how training errors are to be corrected.

Dorigo and Colombetti (1998) use decomposition of tasks by a trainer to "shape" robot behaviour. We also make use of this idea however we do not use either evolutionary or reinforcement learning techniques to create or modify robot behaviour.

Learning policies from demonstrated trajectories seems to be appropriate when the goals of the task are known and the task itself is self-contained. For example when learning to duplicate human movements (Ijspeert, Nakanishi and Schaal 2002) or play tennis strokes (Miyamoto and Kawato 1998) the goal of the task is already known or programmed into the learning mechanism. It is made explicit by the programmer for the specific (although mechanically complex) task to be solved. It is difficult to see how a new task could be incorporated into the existing learned policy without further explicit programming. However, the ability of these systems to reproduce and generalise complex movements is impressive. We see complex tasks such as these as being functions available within a training system which a human teacher could then exploit to further train the robot.

Demiris and Hayes (2002) and Johnson and Demiris (2005) use a *model based policy learning* approach in their work where coupled inverse and forward models imitate observed actions. The prediction capability of the forward models are matched against the current observed and geometrically transformed state of the model in order to weight a set of inverse models paired with their forward component. The inverse model with the highest weighting is then chosen as candidate for the imitation process. The innovative features of this approach are the predictive attributes of the forward models and the fact that it has functional similarities to the biological mechanisms of mirror neurons found in the brains of monkeys (Gallese and Goldman 1998). Our models also apply this predictive idea on actions and we suggest that it is a step towards true observational learning.

Bentivenga and Atkeson (2002) also use *model based policy learning* to construct a learning framework using a memory-based approach. Physical and simulated robots learn to play games of "marble maze" by recording exteroceptive data (ball angle/velocity, board tilt angles) and primitive type (roll ball away from corner, roll ball off wall) from a human demonstrator. The robot is able to select the appropriate primitive by analysing a memory model to find the nearest example to the current state. Parameters for the primitive are constructed using locally weighted regression on points nearest the selected primitive. This technique is also related to loose-perceptual matching methods (Alissandrakis, Nehaniv, Dautenhahn and Saunders 2005).

Memory based learning approaches such as described above have a number of technical advantages. Firstly, complex functions can be learned by focusing on sets of less complex local approximations. Secondly, the local approximation for the target query is based on the training data at the time of the query and not on a pre-built function approximation. This means that additional training instances can be added immediately without the need to rebuild a target function (which would be the case for an inductive or neural network approach).

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 and production rule approaches to learning by demonstration

[2] *There are a number of machine learning techniques that have been used for robot learning. Mahadevan (1996) provides a useful comparison of some of these techniques including inductive, explanation based, reinforcement and evolutionary learning paradigms .*

where both good and bad training experiences are stored with equal weight and are not easily corrected.

5.2 Approach

It is noticeable that many of the examples described above have the ability to learn complex tasks based on some form of observation (where observation can be both direct and from post-processing of sensory data). However there are few mechanisms which allow another task to be both learned and included into the repertoire of previously learned functions. Our approach is to provide a framework which will both *learn a particular task* and have the ability to *add this task to an existing control mechanism* as follows:

- *A policy needs to be learned based on the sensory state of system itself* - we avoid the correspondence issue by having the pupil experience the same set of actions and sensory states as the teacher by the process of *putting through*. Thus there is no need for observational matching and similarly there is no correspondence problem as the robot is corresponding directly with itself. The robot learns to associate the actions moulded on its body against perceived environmental state. Currently our robots have no mechanism for proprioceptive feedback, therefore we control the movements of the robot 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 robot's actuators[3]. During the process of putting through the human trainer has no access to the robot's internal state or perceptions and in fact may not be aware of the form of sensorimotor feedback that the robot is experiencing i.e. the trainer may not be aware or have knowledge of the sensor arrangement, sensor types or sensory-motor modalities of the system.
- *The robot needs to be aware of when tasks have a specific goal* - we make this an explicit part of the training sequence.
- *Learning must be carried out in real-time and be subsequently modified or enhanced with additional learning experiences* – this is made possible by using memory based learning methods.
- *The new learning experience should not corrupt other previously learned experiences* - we allow the construction of a hierarchy of memory models to provide this.
- *A facility through which the robot informs the trainer when it already has knowledge of a task should be provided* - we exploit the predictive and bi-directional nature of the memory models to achieve this.
- *Selective sensory attention on the robot is important to ensure that the robot performs effectively* - this is implemented by automatic weighting and selection of the sensory attributes following the training sequence.

One of the key points in addressing many of the issues described is that a teacher constructs an appropriate learning environment for the robot. This idea of scaffolding addresses the problem of choosing the salience of the experienced actions and is enhanced by the algorithmic techniques of *information gain weighting* and *attribute selection* described in the forthcoming sections. We do this while exploiting and extending some of the techniques described by other researchers above in a new framework.

6. Framework

For this study we use a Pioneer P3-DX robot with 16 sonars, 10 bump sensors, a Canon VCC4 Video camera with a pan/tilt unit and a 5-DOF arm.

The robot is taught three tasks. Firstly to point to a patterned object, secondly to track and follow a human carrying that object and finally (for safety reasons) to avoid obstacles. All three tasks are taught iteratively by a human trainer, correcting mistakes and testing the training process throughout (see figure 1).

Fig. 1. The environment showing the Pioneer robot pointing to a patterned object. The robot is equipped with sonars, bump sensors, a pan/tilt camera and 5-DOF arm.

The sensory feedback to the learning mechanism is the proprioceptive state of the arm angles, pan/tilt unit, sonars and bump sensors together with the centre point (XY coordinates) and distance to the centre a chosen

[3] *In some cases the use of tele-operation to control the robot's actuators may be an advantage. For example, for very large, powerful robots where human safety may be an issue.*

patterned object[4]. An object is detected and tracked in the camera frame using the ARToolkit system (Kato 2006). We ensure that the trainer knows when the robot can see the object by placing an arbitrary virtual object over the pattern (see figure 2).

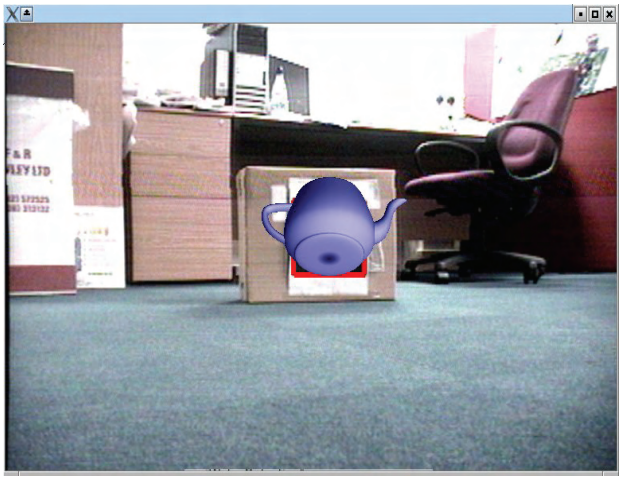

Fig. 2. The patterned object from the robot's point of view. The virtual teapot indicates to the trainer that the robot has correctly identified the pattern.

In the following sections we describe:

1. How the robot uses similarity measures to learn.
2. How *putting through* is used in matching robot states and trainer directed actions.
3. How relevant attributes in the robot state are selected and amplified by scaffolding.
4. How the robot is able to learn new tasks and modify existing tasks.
5. How the robot uses a mechanism of action prediction to inform the trainer when it has an existing competence.
6. How the robot selects what action to take in its current environmental state.
7. How the automatic choice of relevant attributes ensures that the robot is selective in its sensory attentional mechanism.

6.1. Learning Mechanism

We use a memory based ``lazy'' learning method (see Mitchell (1997) for details) 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 sensorimotor 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 robot's actions. We call this collection of states a *memory model*. When the task is executed the robot 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.

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
Distance	Distance of object from camera
Area	Size of object tracking area
Sonar	Values of the 8 front and 8 rear sonars
ObjectAngle	Angle in degrees to nearest object
Bumpers	Values of the 10 bump sensors
BumpFront/ BumpRear	Values of 2 indicator showing state of combined front or rear bumpers.
Arms	Angles of 6 arm joints

Table 1. State Vector

Prior to this study we used both Manhattan (shown below) and Euclidean distance metrics, however there was very little or no difference between them. We therefore chose the computationally simpler Manhattan metric throughout. The distance between the state vector and the training example being the sum of the distances between the features in each, as follows:

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

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 both experimentally and by cross-validation. However the value of k derived from cross-validation has never yielded a value higher than 5 due primarily to the low number of training instances. We make use of two

[4] *We are not attempting to solve any vision problems in this study and therefore we pre-train the system to detect patterned objects before applying our framework*

software systems to provide the kNN functionality. Firstly, *timlb* the Tilburg University Memory Based Learner (Daelemans, Zavrel, van der Sloot and van den Bosch 2004). This system has the advantage of providing a very efficient *k*D-tree based coding structure (see Moore (1991)) for the training examples so as to speed up performance. Secondly, the *Weka* (Witten and Frank 2005) machine learning toolkit is employed. We use this system to provide various cross-validation facilities and as a means to check the performance of the *timbl* system above. Although not described here we also use the *Weka* system to test alternative machine learning schemes (such as inductive and bayesian approaches) within this application framework.

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 forwards 10cm or continuously
Move Backwards	Move backwards 10cm or continuously
Turn Right	Rotate right by 5° or continuously
Turn Left	Rotate left by 5° or continuously
Increase Joint Angle(n)	Increase one of the six joint angle by 5°
Decrease Joint Angle(n)	Decrease one of the six joint angle by 5°
Move Arm (angle vector)	Move the arm to a given position using the six angles (angle vector)

Table 2. Pre-defined Primitives.

6.2 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 robot by remotely moving it through any sequence from a set of pre-defined basic primitives. This set of primitives are basic actions available to the robot such as turn left/right, move forwards/backwards, etc. (see table 2). The human teacher has no access to the internal state of the robot. By manipulating the robot in this manner we also avoid both the problem of observation by the robot of the human actions and of the correspondence problem between the robot and human.

During the robot `putting through' process a snapshot of the robot's 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.

6.3 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, during tracking of an object using the pan/tilt unit the values of the X-Y object tracking values may be of more relevance than the values of the sonars or arm angles.

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:

$$Gain(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v), \quad (2)$$

where S is the collection of training examples, H(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 quantity can be found in Mitchell (1997) and Quinlan (1993). 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, (setting the weight W_i in equation(1) above). This has the effect of either lengthening or shortening the axes in Euclidean space thus reducing the impact of irrelevant state attributes.

$$W_i = Gain(S,i), \quad (3)$$

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 robot must carry out and therefore is able to construct the training environment appropriately so as to ensure that irrelevant features are removed. The modification of the environment 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.

6.4 Learning New Tasks

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

In "learning" mode the robot can be taught new competencies: *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 (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 robot 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. This is similar to the ethological idea of *Fixed Action Patterns* where animals display certain sequences of movements which are independent of environmental stimuli (Tinbergen 1951). 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 competencies 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. Note that when requested to perform a sequence the robot will simply execute the recorded list of competencies taught by the trainer sequentially. It will make no reference to the environmental state. Each primitive when executed by the robot can be run in two further modes - discrete or continuous. In discrete mode the primitive will execute followed immediately by a "stop" instruction. The continuous mode does not issue the "stop" instruction. Discrete mode allows the trainer put the robot though its range of actions step by step. Continuous mode is typically used after training is complete and enables the robot to execute the primitives without the jerkiness caused by the "stop" instructions above.

The *goal-directed task* level differs from a *sequence* in that during training the actions taken by the system will depend on 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 robot 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. In execution mode the task is iterated until the environmental state is close to a goal state and the task will then terminate (also see figure 3).

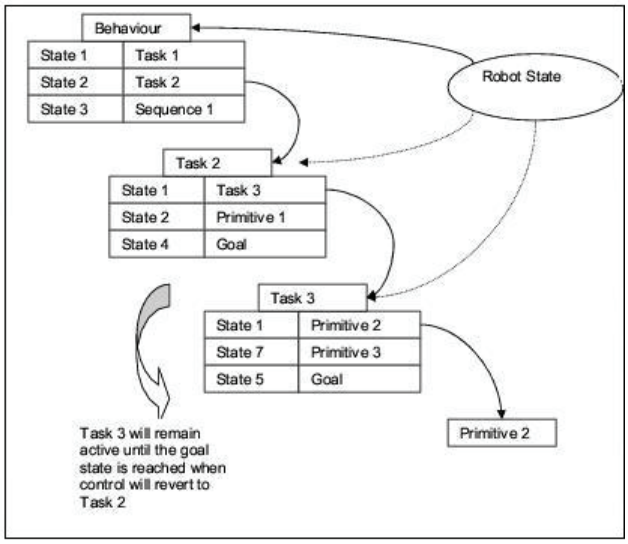

Fig. 3. The schematic illustrates how the current system state is used to select an appropriate action from each memory model. As we proceed down the hierarchy a primitive will eventually be selected for execution. The dotted lines from the system state indicate that each task will only complete once its goal state has been reached. This requires that the system state be polled within each task. Note that each task executed in the hierarchy will only terminate when its goal condition is selected based on the current system state, thus within the lowest selected task the state will be polled after each executed primitive.

For example consider a tracking behaviour, for which the goal state is to have the target centred in the visual field. The trainer would:

1. pan the camera to the right
2. choose the "goal-directed task" level
3. label it (say "trackRight")
4. press the "start learning" button

The robot 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 "trackRight" 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.

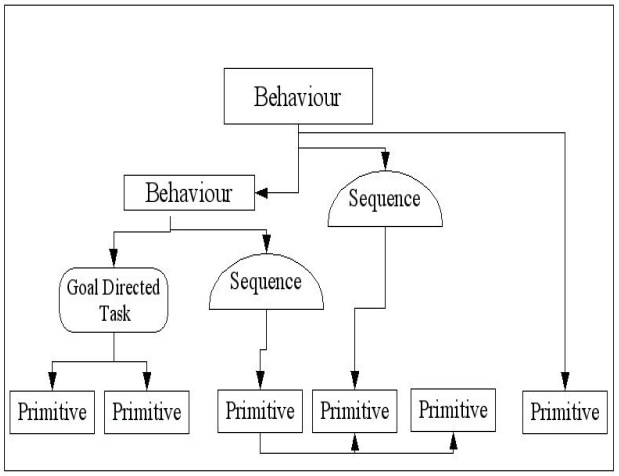

Fig. 4. An example of a trained hierarchy of primitives, primitive sequences, learned goal-directed tasks and behaviours

The *behaviour* level allows the trainer to construct the complete behaviour for the robot 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. A behaviour can also be used by another behaviour, task or sequence as required, the only constraint being that hierarchy must have a behaviour as its topmost node. A key difference between a behaviour and a task is that a task will only yield control to a parent node once its goal state is reached, a behaviour will yield immediately. With careful training the trainer can now build a hierarchy of tasks, sequences and primitives as required (see figure 4).

6.5 Predicting Existing Competencies

Both the goal directed tasks and the behaviours depend 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* 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 confidence factor attached to each memory model that proposed this action is given a reward and all other tasks are penalised. As each training step is taken this cycle is repeated. A bonus is also given to a memory model that has predicted both the current action and has correctly

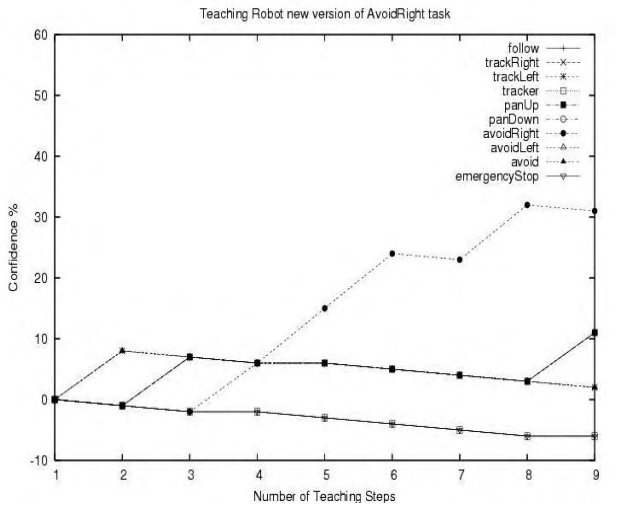

Fig. 5. Recognition of previously learned behaviour using Task Prediction. The graph shows how well each task or behaviour matches the current training sequence. As each training step is taken the task/behaviour most similar to that being trained increases in confidence.

predicted the previous action. This bonus serves to reward consistently correct predictive behaviour. Once the confidence factor of any of the memory models exceeds a global threshold (manually set between 0% and 100% of the confidence factor) the trainer is informed by the robot that it may already have knowledge of this task. Setting a low threshold (nearer to zero) will cause more of the memory models to report possible equivalence; a high threshold will require more matched actions before possible equivalence is reported. For the experiment reported here, the threshold was set at 30% which gave a reasonable compromise between eager and conservative reporting.[5] An example of this process is shown in figure 5. The algorithm is as follows:

$$C_t^N = C_{t-1}^N + \begin{cases} B+R & if\left(A_t^{Taken} = A_t^{\Pr edicted}\right) \\ 0-P & if\left(A_t^{Taken} \neq A_t^{\Pr edicted}\right) \end{cases}, \quad (3)$$

$$B = \begin{cases} 0 & if\left(A_{t-1}^{Taken} \neq A_{t-1}^{\Pr edicted}\right) \\ b & if\left(A_{t-1}^{Taken} = A_{t-1}^{\Pr edicted}\right) \end{cases}, \quad (4)$$

where C^N is the confidence factor proposed for task N at time t, R is the value of the reward, b the value of the bonus and P the value of the penalty. A^{Taken} is the action taken by the trainer at time t, $A^{Predicted}$ is the action predicted by the robot at time t. The value of the reward is set to 100 at the start of the training session, with the penalty and bonus set to the reward divided by the total

[5] *The automatic derivation of this threshold was not attempted but may form part of our future research in this area.*

number of behaviours and tasks in the system. This ensures that the reward is always significantly higher than the bonus but takes account of the fact that there will normally be a large number of incorrect matches for systems with a high number of tasks.

This process is in some ways similar to the prediction capabilities of imitation systems using forward models (Demiris and Hayes 2002; Johnson and Demiris 2005). 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. Our process differs in that the observation is internal rather than external and that actions rather than states are matched.

6.6 Action Selection

The trainer is able to construct a hierarchy of tasks, sequences and primitives and therefore is effectively building an action selection architecture for the robot. At the top behavioural level a decision is made based on the system's current state as to what to execute next (based on the kNN selection). If the selection is a primitive or sequence these will be executed and the next state cycle will begin. Alternatively the selection could be a task or another behaviour. Within the task the system state selects the next appropriate action, which again could be a primitive, sequence, task or behaviour. Working down through the hierarchy eventually results in the execution of a primitive. Precedence of one action over another is entirely based on current environmental state. Precedence of nodes within the hierarchy is controlled by their relative position. Nodes near the top of the hierarchy will have precedence simply because they will be executed before lower level tasks. The stored memory state most similar to the current state is chosen at each level within the framework.

6.7 Selective Attention

The scaffolding process above weights each attribute in the robot's state vector. This technique automatically decides which features of the sensorimotor state vector that are more salient than others for each hierachical component in the overall robot behaviour. We extend this choice not only to the action selection mechanism but also to the sensory selection mechanism. To do this we select attributes only if the information gain value is non-zero. We then poll only those sensors which are part of the new memory model and build the environmental vector attributes to match those in the memory model. This has two effects: first to reduce the number of attributes in the memory model - meaning that the efficiency of the kNN algorithm is enhanced, and secondly by polling only those sensors which provide information defined by this selection to the learning model – meaning that the computational burden of polling unnecessary sensors is reduced. This technique whereby ``cognitive load'' is reduced, may have parallels in our own human experience, for example when first learning to ski we cannot be sure what is important – our physical movements, near objects, far objects, our speed etc. With experience however we focus on only those parts of the task which are relevant, for ski-ing this might be to look into the distance rather than at our ski tips and to concentrate on oncoming obstacles.

7. 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 patterned object held by the trainer. 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. If the trainer moves away from the robot, the robot will follow him/her by keeping track of the object by moving either the pan/tilt unit or its wheels (up/down or moving forward). For safety the robot is also trained to avoid obstacles using its sonars and to carry out an emergency stop and recover procedure if it has bumped into anything as recorded by its bumpers.

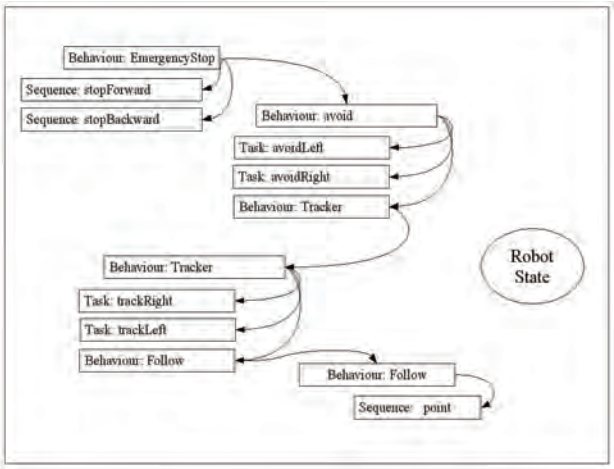

Fig. 6. The derived set of behaviours, tasks and sequences after training.

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 6. Note that for reasons of clarity figures 7-10 show only each unique sequence, task or primitive per memory model. In reality each memory model may have many instances of different states (i.e.

many more rows) in each sequence, behaviour, task or primitive.

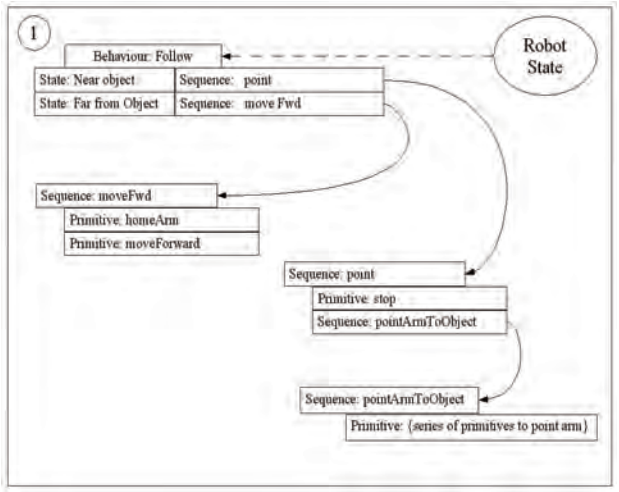

Fig. 7. The follow behaviour created after training.

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, as far as is possible, in a space unaffected by other robots or people. We then construct the learning experience in a series of stages.

Firstly an enhanced primitive is created using the sequencing system. This simply moves the arm to a pointing position. To achieve this a new sequence is created called "point" which moves each of the arm actuators to a particular position. Note that as the actuators are independent, the primitive commands, although issued sequentially, are activated by the motors in parallel.

Secondly, the robot is trained to move forward if the object is outside a given range, the range decided by trainer during training by simply directing the robot to move forward once the object is at a given distance. If closer than this distance the "point" sequence is activated. These steps are shown in figure 7.

The third stage is training the robot to track the object. If the object is in the centre of the robots X-Y tracking plane, then we invoke the "Follow" behaviour taught above. Otherwise we now train the robot to either use its pan/tilt mechanism to track the object in the Y plane (up and down), or if outside its X range to move its wheels left or right, thus bring the object back into view. This is shown in figure 8.

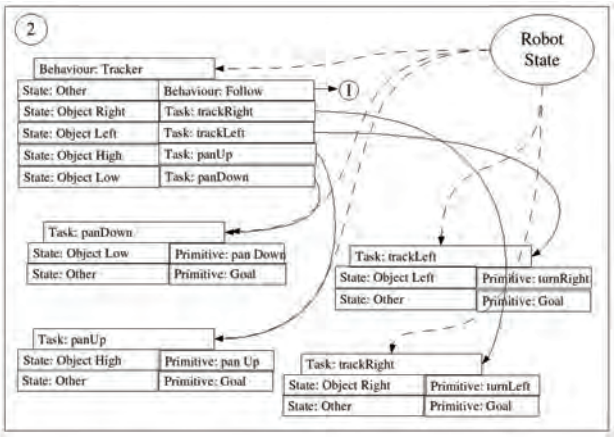

Fig. 8. The tracking behaviour created after training.

Two procedures are necessary for safety. The first is the obstacle avoidance procedure (see figure 9). Here the robot is trained with two goal-oriented tasks to avoid objects on either side of it. The attribute selection mechanism is useful here as only the sonars are relevant, all other sensory feedback is not polled. If the state is not similar to that encountered for avoidance then the trainer directs the robot to carry out the "tracker" behaviour. The final step is an "emergency stop" procedure (see figure 10). The robot is trained to stop and either reverse or go forward if any of the bumpers are activated. This is achieved by physically pressing each bumper on the robot and selecting two pre-taught sequences. The interference picked up during this process by the sonars (as the trainer moves around the robot pressing the bumpers) is ignored by the attribute selection mechanism. The robot is taught to invoke the "avoid" procedure if none of the bumpers are pressed.

During the training process the prediction mechanism is always on. Thus if the trainer were to attempt to train the robot to avoid objects to the right (e.g. create a new version of "avoidRight" in figure 9)), the system would respond with a high similarity measure as further actions are introduced. This is shown in the graph in figure 5 where a new version of avoidRight is trained. As more training steps are taken only the avoidRight task remains at a higher weighting level. The robot informs the trainer of a high match via messages sent to a display box on the control system GUI.

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

8. Performance

For this training example two teachers were used. One teacher holding and moving the patterned object, the other directing the robot actions through the control interface. The time taken to train the robot and number of

training steps taken for this task, which includes time for correcting mistakes and testing the resultant behaviours, is shown in table 3. The table also shows the reduction in the number of sensor attributes that are required to be polled following the application of the attribute selection mechanism.

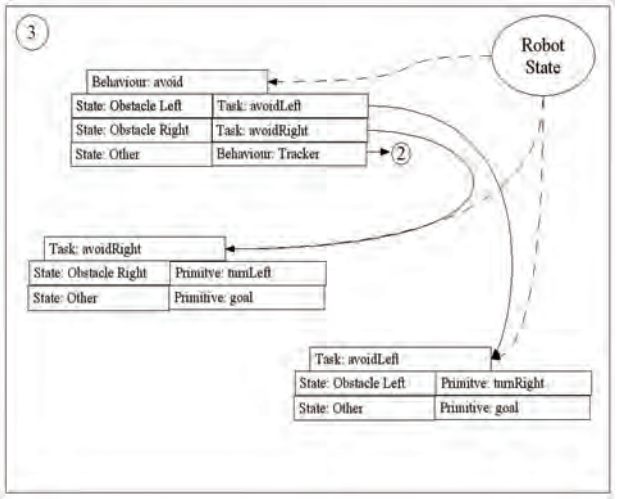

Fig. 9. The avoid behaviour using the sonars.

Training steps. Overall the training took on average 10 steps per item (*SD*=8). However, what is noticeable from the training regime is that for many items very few training steps are required. Typically these are behaviours and sequences. On average each behaviour took 9 steps (*SD*=6) and each task 18 steps (*SD*=2). All sequences took 2 steps. This is because sequences are simple chains of movements and are thus easy to train and test. Behaviours tend to have fewer training instances than goal-directed tasks because the training instances are extreme forms of the underlying task behaviour. For example the avoid behaviour has one instance of a left avoidance situation and one of a right, plus a training instance for when an avoidance situation is not present. The kNN selection algorithm will always choose the trained state nearest to the current state and thus one of the three will be chosen. Tasks contain the main bulk of training as these tend to have to carry out a longer series of actions before a goal state is reached. For example, in the avoid task this would involve moving the robot many times until it was no longer in an obstacle detecting situation.

Overall performance. The performance of the trained robot on the overall behaviour is harder to assess as there are no benchmarks available for comparison. As such the performance of the robot was in part assessed subjectively by the two teachers and in part by objective measures of time required for learning, as well as tested for scalability of behaviour acquisition.

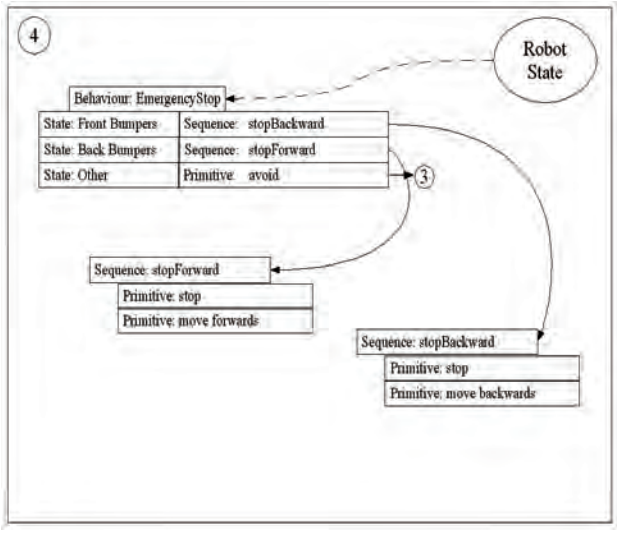

Fig. 10. The emergency stop behaviour based on the bumpers.

In general the robot carried out the task assigned and new behaviours/tasks/sequences could be added dynamically and incrementally upon a growing scaffold of existing ones. However, it was felt that the quality of the goal directed tasks could have been improved with additional training, but as can be seen from table 3 training can take a reasonably long time (in this example over 3 hours for essentially a relatively simple operation). On average each component in the training regime took around 13 minutes to train (*SD*=9), with tasks taking on average 3 minutes longer than behaviours (16 min. vs. 19 min, *SD*=8, *SD*=9 respectively). The time taken to train and test sequences was always around 5 minutes. The times for behaviours and tasks are closer than what might be expected from the analysis of the training steps above, however when each behaviour is tested it inevitably includes the time taken to execute the sub-components.

It should be borne in mind that once learned the relevant components could be re-used in training for additional skills and as such there is probably a higher investment to be made at the earlier stages of learning.

Recognising multiple objects. Although the vision system can be trained to recognise multiple patterns our current methodology will only allow for a single pattern to be recognised in any single camera frame. Thus two patterns cannot be recognised if both exist in the same camera image (although separate patterns can be recognised in separate images). This limits the flexibility of the robot. To obviate this problem would require that each snapshot of the robot's state vector be extended by each pattern. For large numbers of patterns this would become extremely inflexible and severely limit the performance of the system.

Attentional mechanism. The sensory selection mechanism, whereby the number of sensory attributes polled is reduced, gave good results on tasks and most behaviours on average reducing the state vector by 90% of its original size. In one behaviour there was no reduction due to the low number of training instances resulting in the information gain algorithm having very little data to work on. However, the main bulk of sensory polling takes place in the tasks and as such the reduction in load is effective.

Sequence (S), Task (T) or Behaviour (B)	Steps	Size of state vector after attribute selection (out of 41)	Time taken to train, correct and test (mins)
stopBackward (S)	2	n/a	5
stopForward (S)	2	n/a	5
EmergencyStop (B)	12	2	23
Avoid (B)	3	1	10
avoidRight (T)	16	1	27
avoidLeft (T)	15	1	33
Tracker (B)	4	41	8
trackLeft (T)	18	5	16
trackRight (T)	19	5	12
panUp (T)	21	5	11
panDown (T)	20	5	12
Follow (B)	16	5	21
Point (S)	2	n/a	5
PointArmtoObject (S)	2	n/a	5
moveFwd (S)	2	n/a	5
Total for complete behaviour	**154**		**3h 18m**

Table 3. Training time, training steps and sensory attentional selection improvement for the tracking task. Training time includes training, testing and correcting errors.

9. 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 (Saunders, Nehaniv and Dautenhahn 2006) and the second discussed here. The latter architecture has been extended with a predictive capability which aids the teacher in understanding the robot's current capabilities and a facility providing a selective attentional mechanism.

During our experiments we have recognised issues that require further research.

Firstly, the robot has no mechanism for recording historical states i.e. has no form of temporally extended 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 raw camera images. There are also cases where the individual sensor readings do not impart enough information to separate the class value (action) chosen in each memory model. In this case we 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 tend 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. One of the drawbacks of memory based approaches is that with each demonstration the number of training instances increases. The requirement to process all of these instances may be problematic given a very large sensory motor space. For the experiments shown here we use discrete actions and as such the number of training instances were very low (<100 per competence). As stated above the recording of continuous state/action combinations would be preferred and this would inevitably lead to a much larger sensory motor space. This issue of large state spaces could be partly obviated by the use of the attentional mechanisms described above which would serve to reduce the dimensionality of the data combined with efficient processing algorithms such as *k*D-trees (Moore 1991).

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. This implies that the memory model should be separated into parts which are relevant at the given time and may indicate that the training needs to be further scaffolded. Equally, continuous identification of a relevance criterion for the whole stream may need to be extracted in advance, an approach employed by Calinon *et al.* (2006).

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). One could imagine in this situation that the mechanism for task prediction, currently used to inform the trainer of existing competencies, would be used to recognise human actions and offer help e.g. "I see that you are loading the dishwasher, I know how to do that, shall I do it for you?".

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 that would serve, as Prinz puts it,

"to select a certain act, given an intention to achieve certain effect'" (Prinz 2005 [p.143]).

The inverse is the similarity of motor actions informing perception and therefore

"one is to expect certain effects given certain acts" (Prinz 2005 [p.143]).

This idea of the bidirectional nature of perception and action in social learning 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 (Haruno, Wolpert and Kawato 2001; Demiris and Hayes 2002). We intend to pursue this extension in further research using this architectural model.

10. Acknowledgments

The work described in this paper was partially conducted within the EU Integrated Project COGNIRON ("The Cognitive Robot Companion") and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

11. References

Alissandrakis, A.;Nehaniv, C. L.;Dautenhahn, K. and Saunders, J. (2005). "An Approach for Programming Robots by Demonstration: Generalization Across Different Initial Configurations of Manipulated Objects." *Robotic and Autonomous Systems - Special Issue on Robot Learning from Demonstration (to appear)*.

Arentz, D. (1994). The Effect of Disturbances in Behavioural Cloning, Computer Engineering Thesis, School of Computer Science and Engineering, University of New South Wales.

Asada, H. and Izumi, H. (1989). "Automatic Program Generation from Teaching Data for the Hybrid Control of Robots." *IEEE Transactions on Robotics and Autonomation* 5(2): 166-173.

Bentivegna, D. C. and Atkeson, C. G. (2002). A Framework for learning from observation using primitives. Proc. RoboCup Int. Symp., Japan, pp. 263-270, 2002.

Billard, A. and Dautenhahn, K. (1999). "Experiments in social robotics: grounding and use of communication in autonomous agents." *Adaptive Behaviour Journal, Special Issue on Simulation Models of Social Agents* 7(3-4): 415-438.

Brass, M. and Heyes, C. M. (2005). "Imitation: is cognitive neuroscience solving the correspondence problem?" *Trends in Cognitive Science* 9: 489-485.

Byrne, R. W. (1995). *The Thinking Ape: Evolutionary Origins of Intelligence*, Oxford University Press.

Calinon, S.;Guenter, F. and Billard, A. (2006). On learning the statistical representations of a task and generalising it to various contexts. ICRA 2006, pp. 2978-2983, 2006, IEEE.

Caro, T. M. (1980). "Predatory behaviour in domestic cat mothers." *Behaviour* 74: 128-47.

Caro, T. M. and Hauser, M. D. (1992). "Is there teaching in nonhuman animals?" *Quarterly Review of Biology* 67: 151-174.

Daelemans, W.;Zavrel, J.;van der Sloot, K. and van den Bosch, A. (2004). "TiMBL:Tilburg memory-Based Learner." Available from http://ilk.uvt.nl/, Accessed: October 2005,

Demiris, J. and Hayes, G. (2002). Imitation as a Dual-Route Process Featuring Predictive and Learning Components: A Biologically-Plausible Computational Model, In: *Imitation in Animals and Artifacts*, K. Dautenhahn and C. L. Nehaniv, (Ed.), pp. 327-361, MIT Press,

Dillmann, R. (2004). "Teaching and learning of robot tasks via observation of human performance." *Robotics and Autonomous Systems* 47: 109-116.

Dorigo, M. and Colombetti, M. (1998). *Robot Shaping: an experiment in behavior engineering*, MIT Press.

Fikes, R. E. and Nilsson, N. J. (1971). "Strips: a new approach to the application of theorem proving to problem solving." *Artificial Intelligence* 2: 189-208.

Fouts, R. S.;Fouts, D. H. and Van Cantfort, T. E. (1989). The infant Loulis learns signs from cross fostered chimpanzees, In: *Teaching sign language to chimpanzees*, R.A.Gardner, B.T.Gardner and T. E. Van Cantfort, (Ed.), pp. 280-92, State University of New York Press, New York

Gallese, V. and Goldman, A. (1998). "Mirror neurons and the simulation theory of mind-reading." *Trends in Cognitive Sciences* 2(12): 493-501.

Haruno, M.;Wolpert, D. M. and Kawato, M. (2001). "MOSAIC Model for Sensorimotor Learning and Control." *Neural Computation* 13: 2201-2220.

Hayes, G. and Demiris, J. (1994). A robot controller using learning by imitation. Proc. Int. Symp. Intelligent Robotic Systems, Grenoble, 1994, pp. 198-204, 1994.

Herman, L. M. (2002). Vocal, social, and self imitation by bottlenosed dolphins, In: *Imitation in Animals and*

Artifacts, K. Dautenhahn and C. L. Nehaniv, (Ed.), pp. 63-108, MIT Press,

Heyes, C. M. (2002). Transformational and Associative Theories of Imitation, In: *Imitation in Animals and Artifacts*, K. Dautenhahn and C. L. Nehaniv, (Ed.), pp. 501-523, MIT Press,

Ijspeert, J. A.;Nakanishi, J. and Schaal, S. (2002). Movement Imitation with nonlinear dynamical systems in humanoid robots. IEEE International Conference Robotics and Automation, pp. 1398-1403, Washington, 2002, IEEE.

Johnson, M. and Demiris, Y. (2005). Hierarchies of Coupled Inverse and Forward Models for Abstraction in Robot Action Planning, Recognition and Imitation. Proceedings of 3rd International Symposium on Animals and Artifacts at AISB 2005, pp. 69-76, April 2005, The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

k-Team. (2006). "Khepera Robot." Available from www.k-team.com, Accessed: July, 2006

Kaiser, M. and Dillman, R. (1996). Building Elementary Robot Skills from Human Demonstration. IEEE International Conference Robotics and Automation, pp. 2700-2705, 1996, IEEE.

Kaiser, M.;Klingspor, V.;Millán, J. d. R.;Accame, M.;Wallner, F. and Dillmann, R. (1995). "Using Machine Learning Techniques in Real-World Mobile Robots." *IEEE Expert: Intelligent Systems and Their Applications* 10(2): 37-450885-9000.

Kato, H. (2006). "ARToolKit." Available from http://www.hitl.washington.edu/artoolkit/, Accessed: June 2006,

Levas, A. and Selfridge, M. (1984). A user-friendly high-level robot teaching system. IEEE International Conference on Robotics and Automation, pp. 413-416, 1984.

Mahadevan, S. (1996). Machine learning for robots: A comparison of different paradigms. Workshop on Towards Real Autonomy , IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-96), pp. n/a, Osaka, Japan, 1996.

Meltzoff, A. N. (2002). *Imitation as a mechanism of social cognition: Origins of empathy, theory of mind, and the representation of action*, Blackwell Publishers.

Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill International.

Miyamoto, H. and Kawato, M. (1998). "A tennis serve and upswing learning robot based on bi-directional theory." *Neural Networks* 11: 1331-1344.

MobileRobots. (2006). "Pioneer Robot." Available from http://www.activrobots.com/, Accessed: July, 2006

Moore, A. W. (1991). Efficient memory-based learning for robot control,. PhD Dissertation, University of Cambridge, UK.

Moore, B. R. (1996). The Evolution of Imitative Learning, In: *Social Learning in Animals: The Roots of Culture*, C. M. Heyes and B. G. Galef, Jr. , (Ed.), pp. 245-265, Academic Press Inc., 0-12-273965-5, London

Nehaniv, C. L. and Dautenhahn, K. (2002). The Correspondence Problem, In: *Imitation in Animals and Artifacts*, K. Dautenhahn and C. L. Nehaniv, (Ed.), pp. 41-61, MIT Press, Cambridge, Massachusetts

Nicolescu, M. N. and Mataric, M. M. (2001). "Learning and Interacting in Human-Robot Domains." *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 31(5): 419-430.

Piaget, J. (1945/1962). *Play, Dreams, and Imitation in Childhood*, Norton, New York.

Prinz, W. (2005). An Ideomotor Approach to Imitation, In: *Perspectives on Imitation*, N. Chater, (Ed.), pp. 141-156, MIT Press,

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann,San Mateo, CA.

Sammut, C.;Hurst, S.;Kedzier, D. and Michie, D. (1992). Learning to fly. Proc. Ninth Int. Conf. on Machine Learning, pp. 385-393, 1992, Morgan Kaufmann.

Saunders, J. (2006). "Joe Saunders." Available from http://homepages.feis.herts.ac.uk/~sj2ay/, Accessed: July, 17, 2006

Saunders, J.;Nehaniv, C. L. and Dautenhahn, K. (2004). An Experimental Comparison of Imitation Paradigms used in Social Robotics. Proc. IEEE Robot and Human Interactive Communication (ROMAN '04), pp. 691-696, Kurashiki, Japan, September 2004, IEEE Press.

Saunders, J.;Nehaniv, C. L. and Dautenhahn, K. (2005). An Examination of the Static to Dynamic Imitation Spectrum. Proc. 3rd Int. Symp. on Animals and Artifacts at AISB 2005, pp. 109-118, 2005.

Saunders, J.;Nehaniv, C. L. and Dautenhahn, K. (2006). Teaching Robots by Moulding Behavior and Scaffolding the Environment. Proceedings of the ACM Conference on Human-Robot Interaction (HRI06), pp. 118-125, Salt Lake City, USA, March 2006, ACM Press.

Schaal, S.;Ijspeert, A. and Billard, A. (2003). "Computational Approaches to Motor Learning by Imitation." *Philosophical Transactions of the Royal Society of London* Series B, Biological Sciences(358): 537-547.

Tinbergen, N. (1951). *The Study of Instinct*, Clarendon Press.

van Lent, M. and Laird, J. E. (2001). Learning procedural knowledge through observation. Proc. 1st Int. Conf. Knowledge Capture (K-CAP 2001), Victoria, Canada, pp. 179-186, Victoria, British Columbia, Canada, 2001, ACM Press.

Wertsch, J. V. (1985). *Vygotsky and the Social Formation of the Mind*, Harvard University Press.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufman,San Francisco.

Wohlschlager, A.;Gattis, M. and Bekkering, H. (2003). "Action generation and action perception in imitation: an instance of the ideomotor principle." *Philosophical Transactions of the Royal Society* (358): 501-515.