Towards a Mirror System for the Development of Socially-Mediated Skills

Yuval Marom, George Maistros and Gillian Hayes
Institute of Perception, Action, and Behaviour,
Division of Informatics, University of Edinburgh,
5 Forrest Hill, Edinburgh, EH1 2QL, UK
{yuvalm, georgem, gmh}@dai.ed.ac.uk

Abstract

We present a system that attempts to model the functional role of mirror neurons, namely the activation of structures in response to both the observation of a demonstrated task, and its generation. Through social situatedness and a set of innate skills, perceptual and motor structures develop for recognition and reproduction of demonstrated actions. We believe this is an implementation towards a mirror system, and we test it on two platforms, one in simulation involving imitation of object interactions, the second on a physical robot learning from a human to follow walls.

1. Introduction

Epigenetic robotic systems should contain components that are capable of being shaped by the interactions of an agent with its built environment as well as by the social interactions between agents. Our work deals with a particular social learning model that incorporates learning by imitation, and temporal attention. The experience of a learner robot is governed by the actions of a demonstrator, so that the learner gets to sample only those particular parts of the perceptual space pertaining to the skills the teacher is demonstrating. Thus, the social situatedness of the learner and teacher crucially influences the structures that develop within the learner's 'brain'.

We are inspired from the tight coupling between perception and motor control found in mirror neurons. Mirror neurons were found in the *macaque* monkey brain and they were shown to have both visual and motor properties. In fact, single neuron studies by Gallese et al. (1996) and Rizzolatti et al. (1996) explored further the properties of mirror neurons and exposed a strong relationship between perception and motor control. Mirror neurons fire *both* when the monkey performs an action *and* when it observes another monkey or the experimenter perform that same action. Based on these properties it is believed that such a mirror system may form the fundamental basis for imitation (Rizzolatti et al., 2000).

The architecture presented in this paper, shown in Figure 1, is an attempt to model the mirror system, i.e. the functional role of mirror neurons. However, there is no evidence from Neuroscience on how the mirror system is learnt or built. Thus, rather than arbitrarily building a mirror system *a priori*, we use a machine learning approach for this purpose, itself inspired from Psychology and Biology.

Initially the mirror system contains no structures, and we believe that a socially situated agent can develop such structures from observation of a demonstrated action, and then utilise them to reproduce that action. Our architecture relies on the existence of simple innate reactive skills for motor control. These are merely responsible for the inverse kinematics of the physical system, handled by an inverse model.

2. Mirror System

As shown in Figure 1, the mirror system is formed by the coupling of perceptual structures (nodes) and motor structures (schemas). The input to the mirror system is a continuous perceptual stimulus that involves both the observed demonstrator and the imitator's perceived physical environment (*e.g.* objects and walls). The output of the mirror system expresses the desired target state of the agent (*e.g.* postural targets) and goes directly to the motor system for execution.

Each perceptual structure is used to recognise a temporal chunk of the perception of the agent, and is hard-wired to a single motor structure that holds motor targets. The targets can potentially achieve the recognised part of the action, using the innate skills stored in the inverse model. The coupling between a perceptual and motor structure expresses the ability of the agent to perform a particular part of the recognised action. Currently, these associations follow a one-to-one relationship, and in on-going work

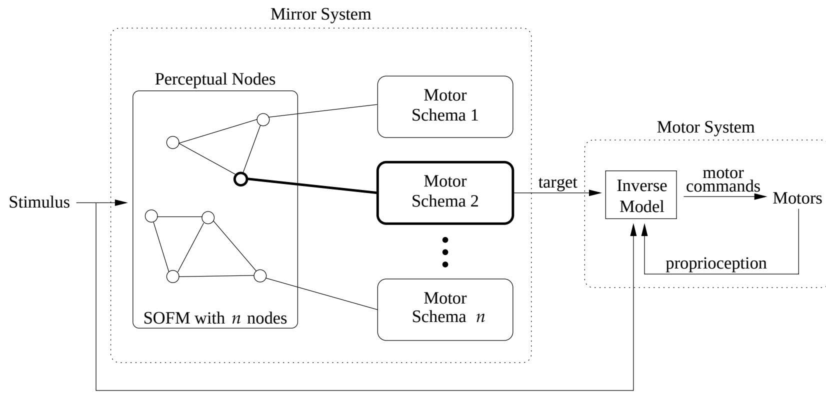


Figure 1: The architecture consists of (1) a mirror system, which is a coupling of perceptual and motor structures that are built up from experience; and (2) a motor system, which consists of innate skills that can convert the output of the mirror system into motor commands.

we are considering more complex relationships (*e.g.* many-to-one).

2.1 Building the Mirror System

The structures that make up the mirror system shown in Figure 1 are built up from experience, during a *learning phase*. A temporal attention system is used to categorise the perceptual input of the mirror system into discrete perceptual structures, and each perceptual structure is then associated directly to a motor structure. The development of the mirror system is therefore perceptuo-centrically driven, *i.e.* it is built bottom-up through *perceptual* experience.

Perceptual Nodes

The perceptual categorisation is achieved using a Self Organising Feature Map (SOFM), which is a useful tool for modelling robotic sensory input (see for example Nehmzow, 1999). The SOFM attempts to cover the sensory input space with a network of nodes, and edges connecting neighbouring nodes determined by a Euclidean distance measure; it is topology-preserving, *i.e.* a cluster of nodes represents a region in the sensory space. We are interested in a variation of the SOFM, where structures (nodes in the network) grow from experience as required, rather than being specified a-priori.

We have adopted and suited to our purposes an algorithm developed by Marsland et al. (2001), which incorporates notions of habituation, novelty detection, and forgetting. Because of the growing, self-organised nature of the system, it reflects at any one time the current perceptual 'memory' of the agent, and can easily adapt and accommodate new experiences.

The attention system is described in detail in (Marom et al., 2002), including a discussion of the various parameters. Briefly, the algorithm involves creating, modifying, and deleting nodes and edges in response to on-line input, as follows:

- the sensory input is converted into a multi-dimensional vector in the same space as the nodes in the SOFM.
- the similarity of the input to all the existing nodes is measured using a Euclidean distance measure, and the closest node is referred to as the 'winning' node;
- if the input matches the winning node well (signalled through a novelty threshold), the winning node and its neighbours habituate, and move towards the input by a small fraction of the distance to the input;
- otherwise the input is novel, so a new node is created between the input and the winning node;
- if a node is completely habituated (signalled through a full-habituation threshold), it is 'frozen': the node does not move from where it is, and cannot be deleted; a forgetting mechanism forces nodes to dishabituate at regular intervals, and hence re-attend to their respective inputs;
- an edge is created between the winning node and the second-best node, while other edges connected to the winning node are aged; when an edge is old enough it is deleted, and any disconnected nodes are deleted.

The system can thus handle novelty, avoid attending to familiar stimuli, but adapt to changing stimuli. The system is said to be attentive when nodes are responding to stimuli, that is, when the nodes are not all fully habituated. There are a number of parameters needed for the algorithm, but the most important one for the experiments in this paper is the novelty threshold, which controls how many nodes are used (the level of granularity in the representation).

In previous work (Marom and Hayes, 2001a), the attention system was used only as a trigger for learning perception-action mappings through a feed-forward neural network with back-propagation. When attentive, the attention system simply provided the trigger for learning, while a completely separate system handled the perception-action associations.

In the work presented here the attention system forms a vital part of perception-action coupling, because the SOFM nodes are associated directly with motor structures. When a new node is created, a motor structure is hard-wired to it, and remains associated with that node thereafter: when the node is updated in response to a stimulus, the motor structure is also updated in response to that stimulus, and when the node is deleted, so is the motor structure.

Motor Schemas

Inspired from Arbib's *Schema Theory* (Arbib, 1981), we use motor schemas to represent the motor structures of the mirror system. Previous work employs both perceptual and motor schemas (Maistros and Hayes, 2001), here however perceptual schemas are replaced by the SOFM nodes. As mentioned above, the motor schemas in the current implementation are created and updated together with SOFM nodes.

The information that is stored in the motor schemas is the motoric representation of the recognised action. One way to obtain this representation is to convert the perceptual information of the demonstrator's action to information that the imitator's own motor system can use. This is the difficult robotics problem of transforming the perceptual space to the motoric space. However, in our implementation we can bypass this problem, because our setup involves the imitator perceiving information that is already in terms of its own body, as will become evident in the experiments below. Therefore, perceived information can be incorporated directly into motor schemas, and we refer to this information as targets to be achieved by the motor system.

A schema update mechanism is responsible for incorporating the perceived information into the motor schema. This update mechanism is very crucial to the ability of the mirror system to generalise and reproduce actions. According to Schema Theory, the type of information that can go inside motor schemas is arbitrary; for instance, sequential targets for a chunk of a movement, parameters for force control, etc. The update mechanism therefore needs to be designed carefully to deal with the chosen representation (for example, processing of temporal information). We are addressing this problem in on-going work, for example the utilisation of a sequence of motor targets in each schema with heuristics to update them (Marom et al., 2002).

In the work presented here we have found that instead of storing a sequence of motor targets in each schema, a *single* representative target is sufficient for modelling the nature and complexities of our tasks. Such a representative is already available in the system through the SOFM node vector, since it generalises over the perceptual space which includes the demonstrated movement. The discussion of the experiments will explain why this approach works for these tasks.

2.2 Using the Mirror System

As described above, the structures of the mirror system are built up from experience. Once built, the mirror system is used in a *recall phase* as follows: the system receives continuous perceptual input; the SOFM is used to recognise that input; the winning SOFM node activates the hard-wired motor schema; this schema provides the output of the mirror system, *i.e.* a motor target; this target is then sent to the motor system for execution.

Notice that the mirror system is only activated perceptually, whereas real mirror neurons can be activated both perceptually and motorically. This issue will be discussed further in the discussion.

3. Motor System

The main component of the motor system is an *inverse model*, used to translate motor targets into motor commands, as used in the control literature. It is a mechanism which, given the robot's current state (perceptual information and proprioceptive feedback) and a desired target state (*e.g.* joint-angle targets), calculates the motor commands that best achieve the desired state. This implementation of the inverse model is a simplified version of the one used by Demiris (1999), who called his inverse models 'behaviours' because they were able to adapt their parameters through a proprioceptive error signal.

In our current implementation the inverse model is innate and remains fixed, and we believe that assuming the existence of such a model is *not* biologically unreasonable. Experiments on early infancy illustrate that prior to the development of advanced motor skills (*i.e.* intentional coordinated goal-directed movements), there is already some basic knowledge about fundamental motor control (Meltzoff and Moore, 1989). One can think of the inverse model as the information about how the robot can use its actuators, say how to use its hands, and the immediate consequences of their use.

By coupling the mirror system with an inverse model, an agent can learn how to use its innate motor skills to achieve a particular demonstrated task. Thus, the ability to recognise and reproduce actions can develop from innate skills through social situatedness. We will demonstrate the implementation of this system on two sets of experiments, a simulated humanoid robot learning to pick up a glass, and a physical robot learning from a human to follow walls. We have also implemented the latter in simulation (Marom et al., 2002) but do not include it in this paper due to space considerations.

4. Object-Interactions Experiment

The experiment presented in this section involves two eleven degrees of freedom simulated humanoid robots (waist upwards), a demonstrator and an imitator, interacting with an object. Each robot has three degrees of freedom at the neck, three at each shoulder, and one at each elbow. The robots are allowed to interact with one object each. The objects are identical and have six degrees of freedom, *i.e.* they can move in any position and orientation in 3D space. The dynamics of each robot are simulated in DynaMechs (McMillan et al., 1995), a collection of C++ libraries that simulate the physics involved with objects and joint control. The torque for the control of each joint, *i.e.* the input to DynaMechs, is calculated with the aid of a Proportional-Integral-Derivative (PID) controller, which converts postural targets (*i.e.* via points for each joint) into such torque values.

The demonstrator is controlled by a sequence of such postural targets to interact with its object, which is lying on a surface at waist level. The postural targets control the demonstrator to 'grasp' the object, pick it up, 'drink' its hypothetical contents, and then put it back on the surface (see Figure 2). The absence of fingers, however, as well as software limitations lead to a rather crude robot-object interaction: namely the object is merely attached to (or detached from) the wrist, as long as this is desired and the wrist is close enough, *i.e.* the object is weightless.

The system described in the previous section is used to model and control the perceptual-motor skill *of the imitator*; the skills of the demonstrator are hand-crafted and do not change; this is the case for all the experiments reported in this paper. In this experiment, the input to the system comes from a crude approximation to visual perception which consists of the joint angles of the observed demonstrator (11 degrees of freedom), plus their corresponding joint velocities (another 11), plus the position and orientation information of the observed object (6 degrees of freedom), plus their corresponding velocities (another 6) — 34 dimensions in total, where noise is also added to each. Similarly, the proprioception of the imitator is approximated as explicit access to the noisy version of its own joint-angles and joint-velocities.

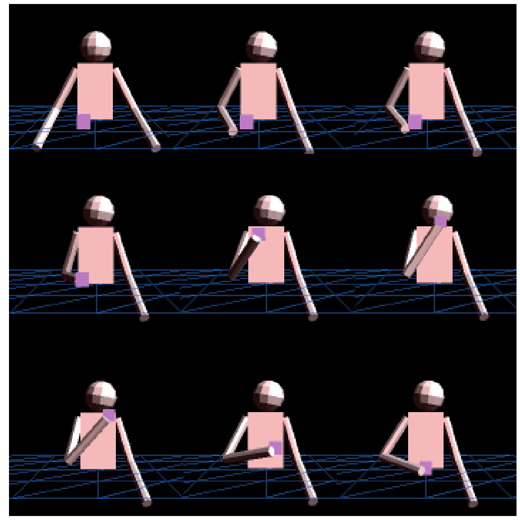


Figure 2: A sequence of snapshots of the demonstrated behaviour; left to right, top to bottom.

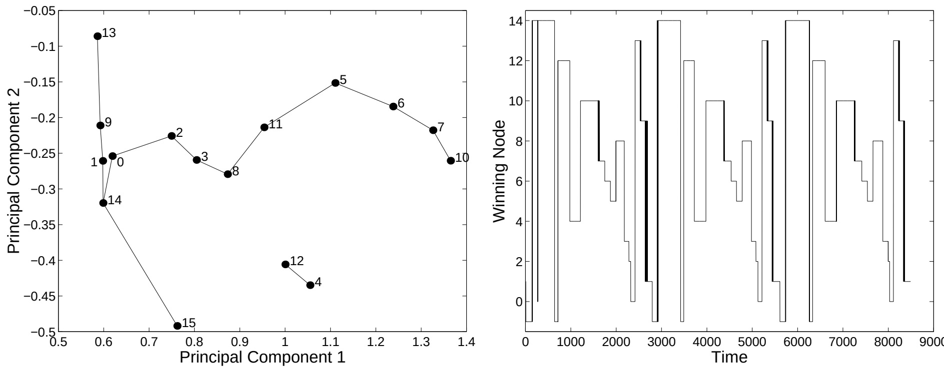
Notice that the imitator perceives the demonstrator in the same way it perceives itself, and thus the imitator can directly represent what it is trying to imitate, as described in Section 2. In other words, the perceived input can be directly stored as targets to be achieved by the motor system (via the inverse model).

Here the inverse model consists of 2 parts: (1) the PID controller is used for posture control: targets are passed into the PID controller which together with proprioceptive feedback calculates the required torque (or motor commands) of each limb; (2) a set of boundary conditions for object-interactions, which specify when the wrist is close enough to pick up the glass, when the wrist/glass is close enough to the mouth to 'drink', and when the wrist/glass is close enough to the table to put down the glass (these boundary conditions are set to a radius of approximately 4 cm from the centre of the glass, mouth, and table).

To summarise, the stimulus in this experiment is a 34-dimensional vector that represents the imitator's perception of the demonstrator and the object, and the motor commands calculated by the inverse model are used to control each of the imitator's limbs, and the object-interaction mechanism.

4.1 Learning & Recall

In this experiment, during the learning phase the imitator merely observes the demonstrator, analysing the visual perception of what the demonstrator is doing by training its SOFM; it does not try to replicate



(a) (b)

Figure 3: The object-interactions experiment: (a) a SOFM produced by the attention system at the end of a complete learning episode, projected onto the first 2 principal components; (b) the SOFM node activation at the recall phase.

the demonstrator's actions. The learning phase consists of 20 demonstration episodes (3000 steps each) of the object-interaction that was described above. Figure 3(a) shows one SOFM network that the system can produce with particular parameter values.

Since the dimensionality of the input space is quite high (34 dimensions), we have used a dimensionality reduction technique called Principal Component Analysis (PCA) to display the SOFM[1] (the principal components used in the figure account for approximately 80% of the variance).

In Figure 3(a) we can distinguish four parts: one fairly straight curve (nodes 15–14), a disjoint cluster (12–4–10), two half loops starting at node 7 and ending at node 0, and another fairly straight curve (1–13). These parts in fact correspond to the four parts of the 'drinking from a glass' behaviour: (i) approach the glass; (ii) pick up, bring to the mouth; (iii) put down; (iv) move away from the glass.

In the recall phase, the structures of the mirror system are fixed; the demonstrator performs the object-interaction again, but now the imitator tries to match this behaviour; the interaction is repeated 3 times (again 3000 steps each). The SOFM receives continuous perceptual input which activates the best-matching node; the corresponding motor schema provides a motor target, which is simply the SOFM node vector; the target is passed to the motor system where motor commands are calculated by the inverse model, to achieve the perceptual state recognised by the SOFM node. If the winning node does not match the input well enough (signalled through the novelty threshold), then no motor commands are produced and the imitator maintains its current posture; this illustrates recognition failure either due to unfamiliar visual perception, insufficient learning, or inability to learn what was demonstrated in the learning phase.

[1]PCA finds the most statistically significant dimensions, called Principal Components, in a multivariate dataset.

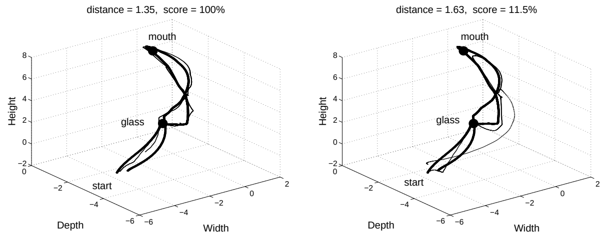
Figure 3(b) shows the SOFM activation during the recall phase, *i.e.* the sequence of nodes that are activated in response to the input, for the SOFM shown in Figure 3(a). A winning node of -1 indicates a poor match and hence no winning node. We observe that the SOFM nodes created at the learning phase, are activated in sequence at the recall phase; node 14 represents grasping the glass; nodes 12–4–10 represent lifting the glass and moving it to the mouth; nodes 7–6–5–8–3–2–0 from the mouth back on to the surface; node 13 away from the glass (towards the starting posture); nodes 9–1 towards the glass once again.

4.2 Results

Figure 4 shows the trajectories of the right-hand wrists of both the demonstrator (in bold font) and the imitator (normal font) in a single episode in the recall phase. Figure 4(a) shows a successfully learned action, *i.e.* the trajectories are close to each other, whereas Figure 4(b) shows a less successful one: the trajectories are further apart. This reflects exactly what we have visually observed: natural motor control with reasonable degree of accuracy imitation in Figure 4(a), and much less accurate in Figure 4(b); in the latter the imitator misses its mouth and does not place the glass back on the table.

As well as visual inspections, we are also evaluating our system numerically on a task-specific basis. In this experiment we use two evaluation measures: a 'distance' measure, which calculates the position of the wrist over time relative to the position of the demonstrator's wrist (*i.e.* the distance between the two curves in the plots of Figure 4), and a 'score' for successful execution of the task (*i.e.* picking glass up, drinking, putting down). The measures are described in more detail below.

The analysis of the results consists of evaluating



(a) (b)

Figure 4: Visual inspection of the recalled behaviour: the trajectories of the right hand wrists of the demonstrator (bold font) and the imitator (normal font) in a single episode in the recall phase. (a) trajectories of a successfully learned behaviour; (b) a less successful one. The black spheres on the plots denote the task subgoals (*i.e.* glass and mouth), and their radii the corresponding boundary conditions.

the performance measures as a function of SOFM network size (number of nodes, which is governed by the novelty threshold). We have used 22 different novelty threshold values, which result in networks of sizes varying from 5 to 60; for each threshold value the experiment is repeated 50 times.

We measure the distance between the two trajectories by calculating the Euclidean distance between them at each time-step (this simple calculation does not take into account the time-lag between imitator and demonstrator, however we have also calculated the distance using a short-term memory window, and the results were similar). The distances measured are shown in Figure 5, as a function of SOFM network size. We see that the path trajectories are consistently close to the demonstrator's, for all network sizes greater than 5.

Note that the distance we are calculating is only a measure of the form of the movement; it does not measure how successful the imitator is in achieving

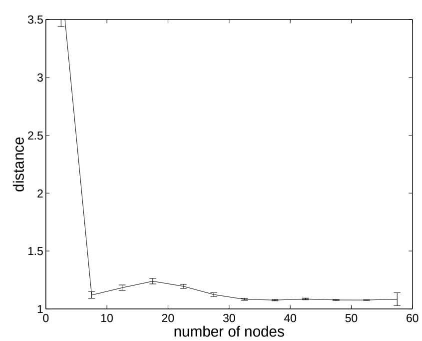


Figure 5: Evaluation of the recalled behaviour as a function of network size. The Euclidean distance between the right-hand wrists of the demonstrator and the imitator is an approximated measure of the form of the movement.

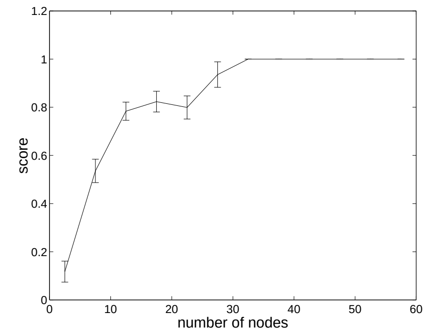


Figure 6: Score obtained at the recall phase, as a function of network size. The score is a measure of how well the imitator achieves the task.

the task. In fact it is possible that the trajectory of the imitator is close to the demonstrator's, but the imitator fails to to pick up the glass, etc. This can sometimes happen for small networks, where the trajectory is good, but because there are not many nodes the imitator is actually 'cutting corners' and missing the glass, or missing the mouth, etc.; that is there is not enough detail in the representation.

To test the success of the recalled action with respect to the task, we have devised another measure which scores the behaviour. The imitator can get scores by achieving any combination of the following 3 goals: (A) picking up the glass, (B) 'drinking' from the glass, and (C) putting the glass back on the table; a bonus is given if all 3 goals are achieved, which corresponds to a perfect execution of the task. We calculate a score similarly for the demonstrator, and then scale the imitator's score by the demonstrator's, because the imitator can only perform as well as the demonstrator, who occasionally fails in parts of the task (due to noise).

The scores obtained are shown in Figure 6, as a function of network size. We see that indeed the small-sized networks (< 10 nodes) that achieve good trajectories (Figure 5), are in fact not very successfully in achieving the task. The best networks are those of 30 or more nodes, as they can match the demonstrated movement *and* achieve perfect object-interactions.

5. Wall-Following Experiment

The experiments presented in this section were performed using our Real World Interface (RWI) B21 robot, Gillespie, and a human demonstrator, as shown in Figure 7; the robot is programmed to detect and follow the human using its on-board video camera; this is done using a simple colour-tracking algorithm — the demonstrator is wearing a green shirt which is easily detectable. The arena is approximately a 5m $\times$ 5m square. The task is wall-following.

Figure 7: The robot environment in the wall-following experiment. The robot is programmed to track and follow a human demonstrator using its on-board camera; the input to the mirror system comes from the sonar sensors around its body.

The input to the system comes from 20 sonar sensors around the top of the robot, which in practice are not affected by the presence of the demonstrator.

The learner can sometimes lose the demonstrator, so it only inspects its perceptual input when the demonstrator is in sight, that is, when it is in a social context. Otherwise, the attention system would encounter situations not relevant to the task (Marom and Hayes, 2001b). We regard this setup as social situatedness in the sense that information is implicitly shared between the demonstrator and imitator about the specific task to be learned. This is an appealing idea that has been experimented with before (for example Hayes and Demiris, 1994; Billard and Dautenhahn, 1997). Further, this setup allows for the perceived input to be represented directly as targets to be achieved by the motor system (through the inverse model), as described in Section 2. The object-interactions in this case correspond to how the robot responds to being near a wall or away from it.

In this experiment the inverse model is not as straightforward as in the first experiment where the PID provides an intuitive inverse model. In similar experiments in simulation we have used an inverse model which consisted of a discretised database of states and transition matrices obtained by letting the agent explore its environment (Marom et al., 2002). We have found that in the physical system it is difficult to obtain such an inverse model which is reliable; we believe that a more sophisticated approach is required, such as reinforcement learning, and leave that to further work. We overcame this problem by handcrafting a set of rules that operate on a small set of states which reliably generalise the robot's state space, and with which the robot can decide how to get from one perceptual state to another.

To summarise, the stimulus in this experiment is a 20-dimensional vector that represents the robot's perception of the wall, and the motor commands calculated by the inverse model are used to control the robot's motors to move forward, turn left, or turn right.

5.1 Learning & Recall

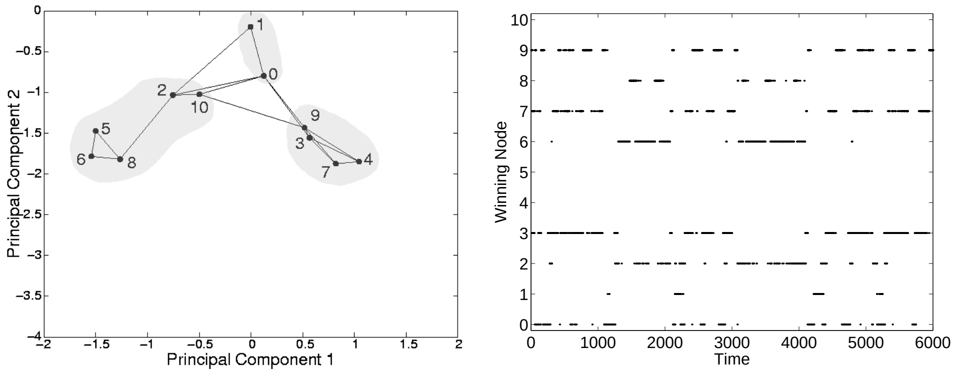
In the learning phase the robot follows behind the human demonstrator for 10000 steps, which is approximately 40 minutes of real time; due to hardware and practical limitations, all the information is stored for off-board learning. The various recall runs reported below are all based on this one dataset.

Figure 8(a) shows one SOFM network that the attention system can produce and highlights the emergent clusters in the SOFM. As in the previous experiment, since the dimensionality of the sensor space is too high to visualise, we have used PCA to reduce the number of dimensions to two (the principal components used in the figure account for approximately 70% of the variance). We can see a cluster for no-wall (at the top), and as we move away from it we move towards clusters corresponding to the walls (left and right).

Following training at the learning phase, the mirror system is fixed and is used to control and test the robot, as in the previous experiment: the robot is placed in the environment on its own; at each step the robot's perception activates one of the nodes in its SOFM; the corresponding motor schema provides a motor target which, again, is simply the SOFM node vector; and this target is passed to the inverse model which selects the best action likely to achieve it. When no node is active (the match is very poor) a 'wandering' behaviour is triggered (the robot moves around randomly). The recall phase consists of 6000 steps, which corresponds to around 23 minutes of real time; to avoid the robot following the wall on one side for the duration of the run, and thus not testing the learned behaviour fully, we use an 'interrupt', which forces the robot to turn away from the wall, every 1000 steps during the run.

We have also equipped the robot with a built-in obstacle avoidance behaviour to protect it from unsuccessful learning and also from situations not encountered at the learning phase. For example, when the robot follows behind the demonstrator, it never sees the wall directly in front of it, so we do not expect it to know how to handle such a situation in the recall phase, but we also don't want it to drive into the wall! To account for unsuccessful learning we penalise the evaluation whenever the obstacle-avoidance is triggered.

Figure 8(b) shows the node-activation at the recall phase, of the SOFM shown in Figure 8(a). Firstly, we see that the nodes that form clusters are also activated together and intermittently at the recall phase.



(a) (b)

Figure 8: The wall-following experiment: (a) a SOFM produced by the attention system at the end of a learning episode, projected onto the first 2 principal components; the emergent clusters are highlighted; (b) the SOFM node activation at the recall phase.

For example nodes 3,7, and 9 form a cluster for a wall on one side and are also intermittently active in the recall phase. Secondly, we see an emergent sequence of activations as the robot moves around trying to recall the information from the motor schemas: the activations of nodes for wall perception on either side (*i.e.* nodes 3-7-9, or nodes 2-6-8) are separated by activations of nodes for no-wall perception (*i.e.* nodes 0-1). This reflects our visual observations of the actual behaviour of the robot; for example, Figure 8(b) corresponds to following a right wall, then no wall, then following a right wall again, then no wall, then following a left wall, etc.

Note that the activation dynamics here are different than in the first experiment (Figure 3(b)), because of what is being modelled. In the first experiment the behaviour modelled by the SOFM has a true sense of sequence in it (moving the hand towards a glass, picking it up, etc.); each node stores a part of that sequence. In the wall-following experiment, each node, or rather cluster of nodes, corresponds to being in a particular perceptual 'state', and the motor skills are responsible for *maintaining* that state (for example, fine-tuning to stay next to a wall on the left). For this reason we do not have a situation where one node wins consistently for a long period; rather, the robot will keep (re)adjusting itself by activating alternate nodes within the same cluster.

5.2 Results

As in the first experiment, we are also evaluating the performance of the system on this task numerically; we do this by calculating the 'energy' that the robot acquires from the wall at particular orientations from it. At the end of the recall episode we can look at the accumulated energy as a measure of how well the robot performs the wall-following task. Due to practical time limitations the amount of data available is much smaller than in the first experiment, however we believe it is sufficient for a reliable evaluation. The dataset obtained in the learning phase was used with different values of the novelty threshold to obtain various SOFM networks for testing: around 15 different network sizes were produced, each repeated between 2 and 5 times.

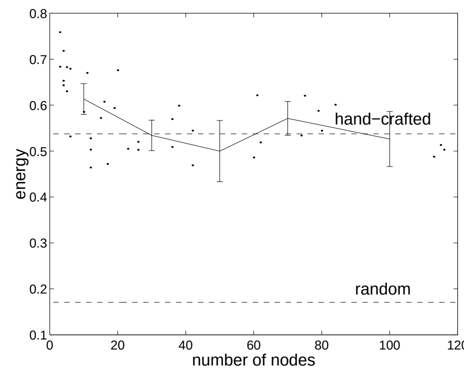


Figure 9: Evaluation of the recalled behaviour as a function of network size. Energy is measured from the sensors at particular configurations from the wall, and compared to energies acquired by a hand-crafted behaviour and a random wandering behaviour. Raw data are also shown.

The energies acquired by the various networks are shown in Figure 9, together with energies acquired by a hand-crafted wall-following behaviour, and a random wandering behaviour. We see that small networks are preferred and that increasing them by more than 30 nodes does not have a significant effect.

6. Discussion

We have presented an architecture that can represent the sensory-motor experiences of a robot, in such a way to enable recognition and reproduction of a demonstrated task. A mirror system is built up from experience using an on-line machine learning approach (variation of the SOFM) that self-organises, and in effect temporally segments the experiences of the robot. We have seen that different SOFM network sizes are needed to represent each of the two tasks presented, and this is because they have differing complexities; the object-interactions experiment requires more nodes than the wall-following one.

Our mirror system represents how to use basic innate skills such as moving a hand to particular positions, attaching a weightless glass to the wrist, etc. in order to perform the task of the first experiment; or turning to face the wall, turning to be parallel to the wall, keeping parallel to the wall, etc., in order to perform the task of the second experiment.

The nature of both tasks involves *only* observable information, *i.e.* there are no physical constraints. Recall that motor schemas hold the SOFM node vector information, which is of a purely perceptual nature. Therefore perceptual information in the structures of the mirror system is alone sufficient to recognise and reproduce both tasks. In future work we intend to devise new tasks that will require the architecture to be extended, so that motor schemas include factors of a more motoric nature; for example, weight, force control, somatosensory feedback, etc.

As briefly discussed in Section 2, our system is only an attempt *towards* an implementation of the biological mirror system, since it is *only* perceptually activated. A more complete implementation would involve the system being activated both perceptually and motorically. This would involve extending our current architecture to include external factors to activate the mirror system motorically; this is currently beyond the scope of our work.

Our architecture relies on the existence of an inverse model (a set of innate skills). This is not biologically unreasonable, because there are low-level motor skills already present at birth, prior to advanced motor control. Further, these skills need not necessarily be hand-coded prior to implementation of the mirror system; they can in fact be acquired and/or modified through self-exploration, as we have done in other experiments (Marom et al., 2002).

We present a system that attempts to model the functional role of mirror neurons, namely the activation of structures in response to both the observation of a demonstrated task, and its generation. Through social situatedness and a set of innate skills, perceptual and motor structures develop for the recognition and reproduction of demonstrated actions. We believe this is an implementation towards a mirror system, and we have tested it on two different platforms.

7. Related Work

Pomplun and Matarić (2000) and Fod et al. (2000) have developed other methods to segment and cluster data from demonstrated movements, using techniques such as Principal Component Analysis and K-Means clustering; these approaches differ from our work in that they operate on a batch of data and hence need to be re-trained to handle novel movements.

Andry et al. (2002) use a similar approach to ours, however they also self-organise the robot's proprioception; a simple winner-take-all mechanism is used to segment the visual input into 'perceptual structures', and then each perceptual structure is associated with a separate SOFM that maps the robot's proprioception. This means that a one-to-many relationship between perception and action is built, as we mention in this paper; *i.e.* there are different ways of achieving each perceptual situation.

There are several examples of implementations that rely on the pre-existence of perceptual-motor structures, *i.e.* where the movement segmentation is performed either off-line, such as the examples mentioned above, or hand-coded arbitrarily by the designer, such as our earlier work (Maistros and Hayes, 2000). Such perceptual-motor structures are often referred to as *primitives* (Demiris and Matarić, 1998; Schaal, 1999; Matarić, 2000).

Although our architecture would need to be extended to include motoric activation of the mirror system, it is generally the case also with other implementations that they lack both types of activation. In contrast, Fagg and Arbib (1998) and Arbib et al. (2000) model the mirror system together with other brain areas including the ones responsible for action selection, constraints, motor plans, etc., which they use to trigger their mirror system motorically.

References

Andry, P., Gaussier, P., and Nadel, J. (2002). From visuo-motor development to low-level imitation. In *Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Edinburgh, UK.

Arbib, M. A. (1981). Perceptual structures and distributed motor control. In Brooks, V. B., (Ed.), *Handbook of Physiology - The Nervous System II. Motor Control*, pages 1449–1480. Amer. Physiol. Soc.

Arbib, M. A., Billard, A., Iacoboni, M., and Oztop, E. (2000). Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks*, 13(8–9):975–997.

Billard, A. and Dautenhahn, K. (1997). Grounding communication in situated, social robots. In *Proceedings of Towards Intelligent Mobile Robots Conference.*

Demiris, J. (1999). *Movement Imitation Mechanisms in Robots and Humans.* PhD thesis, Division of Informatics, University of Edinburgh.

Demiris, J. and Matarić, M. J. (1998). Perceptuo-motor primitives in imitation. In *Working Notes, Autonomous Agents '98 Workshop on Agents In Interaction - Acquiring Competence Through Imitation.*

Fagg, A. H. and Arbib, M. A. (1998). Modeling parietal-premotor interactions in primate control of grasping. *Neural Networks*, 11(7/8):1277–1303.

Fod, A., Matarić, M. J., and Jenkins, O. C. (2000). Automated derivation of primitives for movement classification. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000)*, MIT, Cambridge, MA.

Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119:593–609.

Hayes, G. M. and Demiris, J. (1994). A robot controller using learning by imitation. In Borkowski, A. and Crowley, J. L., (Eds.), *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, pages 198–204. LIFIA-IMAG, Grenoble.

Maistros, G. and Hayes, G. M. (2000). An imitation mechanism inspired from neurophysiology. In *Proceedings of EmerNet: Third International Workshop on Current Computational Architectures Integrating Neural Networks and Neuroscience.* Durham, UK.

Maistros, G. and Hayes, G. M. (2001). An imitation mechanism for goal-directed actions. In Nehmzow, U. and Melhuish, C., (Eds.), *Proceedings of TIMR 2001 – Towards Intelligent Mobile Robots*, Manchester University.

Marom, Y. and Hayes, G. M. (2001a). Attention and social situatedness for skill acquisition. In *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, Lund, Sweden.

Marom, Y. and Hayes, G. M. (2001b). Interacting with a robot to enhance its perceptual attention. In Nehmzow, U. and Melhuish, C., (Eds.), *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Marom, Y., Maistros, G., and Hayes, G. (2002). Experiments with a social learning model. *Adaptive Behavior*, Special Issue 9. In press.

Marsland, S., Nehmzow, U., and Shapiro, J. (2001). Novelty detection in large environments. In Nehmzow, U. and Melhuish, C., (Eds.), *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Matarić, M. J. (2000). Getting humanoids to move and imitate. *IEEE Intelligent Systems*, 14:18–24.

McMillan, S., Orin, D. E., and McGhee, R. (1995). Dynamechs: An object oriented software package for efficient dynamic simulation of underwater robotic vechicles. In Yuh, J., (Ed.), *Underwater Vechicles: Design and Control*, chapter 3, pages 73–98. TSI Press.

Meltzoff, A. N. and Moore, M. K. (1989). Imitation in newborn infants: Exploring the range of gestures imitated and the underlying mechanisms. *Developmental Psychology*, 25(6):954–962.

Nehaniv, C. and Dautenhahn, K. (2000). Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In Demiris, J. and Birk, A., (Eds.), *Learning Robots: An Interdisciplinary Approach.* World Scientific Series in Robotics and Intelligent Systems, Vol. 24.

Nehmzow, U. (1999). *Mobile Robotics: A Practical Introduction.* Springer Verlag. ISBN 1-85233-173-9.

Pomplun, M. and Matarić, M. J. (2000). Evaluation metrics and results of human arm movement imitation. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000)*, MIT, Cambridge MA.

Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141.

Rizzolatti, G., Fogassi, L., and Gallese, V. (2000). Cortical mechanisms subserving object grasping and action recognition: A new view on the cortical motor functions. In Gazzaniga, M., (Ed.), *The New Cognitive Neurosci.*, pages 539–552. MIT Press.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Current Opinion in Neurobiology*, 3:323–242.