Active Imitation

John Demiris and Gillian Hayes
Institute of Perception, Action and Behaviour
Division of Informatics
University of Edinburgh, EH1 2QL, Scotland, UK

johnde@dai.ed.ac.uk, gmh@dai.ed.ac.uk

Introduction

Imitation mechanisms in artificial and biological agents are of great interest mainly for two reasons: from the engineering point of view, they allow the agent to efficiently utilise the knowledge of other agents in its social environment in order to quickly learn how to perform new tasks. From the scientific point of view, these mechanisms are intriguing since they require the integration of information from the visual, memory, and motor systems. A lot of previous successful attempts to utilise human demonstration as a source of teaching signals in robotics (e.g. (Kuniyoshi et al, 1994), (Kaiser & Dillman, 1996), (Demiris et al, 1997)) fall in the category of what we will term as passive imitation architectures. Within this category, there is a module that is responsible for perceiving some features of the demonstrator's actions, linked with a recognition module which processes the incoming features and usually attempts to classify them as examples of a specific action, followed by the motor system module which executes the commands that will reproduce the observed action. A clear separation between the perception of an action and its production can be observed. This distinction has been challenged by neurophysiological and psychological experiments that have indicated that perception and production of an action might be more tightly coupled. This paper introduces a distributed 'active' imitation architecture: it is termed active to denote that the imitator actively generates and internally performs candidate actions while the demonstrator's action is unfolding, rather than attempting to classify it after it is completed.

Background

The architecture that will be described in the next section makes extensive use of the concepts of behaviours and of forward models. A forward model of a controlled object (a 'plant' as it is known in the control literature) is a function which given the current state of the plant and a control command to be applied on it, outputs the predicted next state. Forward models are important in control engineering since they provide predictions on what the results of applying an action will be before they are perceived by the sensors, thus providing the opportunity to correct predicted errors before they occur.

A behaviour is a function that given the current state of the plant and the target goal(s), outputs the control commands that are needed in order to achieve or maintain the goal(s). Target goals can either be implicit or explicit. For example, for a pick-object behaviour, the target object to be picked up must be stated explicitly and fed in the behaviour, while for a head-nodding-yes behaviour, the target goal is already defined implicitly and the current state is enough to determine the motor commands needed to achieve the goal.

The architecture

In this section, the active imitation architecture will be described, starting initialy from how this architecture can be used to generate behaviours, followed by how such an architecture is used in order to understand demonstrated behaviour.

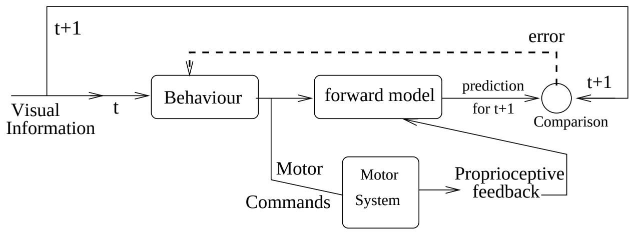

Figure 1: The basic building block - a pairing of a behaviour with a forward model

The basis for this architecture is the behaviour-forward model pairing of figure 1. In order to execute a behaviour with this structure, the behaviour module receives information about the current state (time

t) and the target goal(s), and it outputs the motor commands that it believes are necessary to achieve or maintain the target goal(s). The forward model then provides an estimate of the next state (time $t+1$), and this is fed back to the behaviour, allowing it to adjust any parameters of the behaviour (in our case, it allows the adjustment of the gains of the PID controller used to implement the behaviours (next section), in order to achieve different movement speeds)

More importantly, the same structure can be used in order to match a visually perceived demonstrated behaviour with the imitator's equivalent motor one. This is done by feeding the demonstrator's current state as perceived by the imitator to the behaviour module and having it generate the motor commands that it would output if it was in that state and wanted to execute this particular behaviour. The motor commands are inhibited from being sent to the motor system. The forward model outputs an estimated next state which is a prediction of what the demonstrator's next state will be. This prediction is compared with the actual demonstrator's state at the next time step. This comparison results into an error signal which can be used to increase or decrease the behaviour's confidence value, which is an indicator of how confident the particular imitator's behaviour is that it can match the demonstrated behaviour.

Figure 2 shows the complete architecture which consists of several of the structures that were described above, operating in parallel. When the demonstrator executes a behaviour, the perceived states are fed into the available imitator's behaviours which, in parallel, generate motor commands that are sent to the forward models. The forward models generate predictions about the demonstrator's next state which are compared with the actual demonstrator's state at the next time step, and the error signals resulting from these comparisons affect the confidence values of the behaviours. At the end of the demonstration (or earlier if required) the behaviour with the highest confidence value is selected.

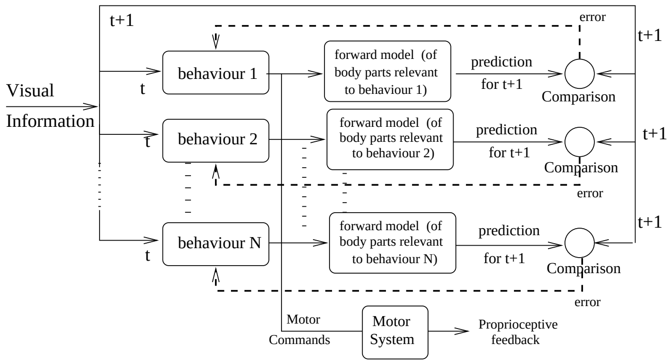

Figure 2: The architecture consists of behaviours paired with forward models

Experiments

In this section we present the results of implementing the architecture above on a dynamics simulator of a thirteen degrees of freedom robot using as a test set the movements that achieve the various letters of the International Standard Semaphore Code (ISSC).

Implementation domain

The ISSC is an alphabet signalling system based on moving the arms to various positions/postures representing each of the letters of the alphabet (an on-line guide to ISSC can be found at *http://www.sacdelta.com/semaphore.htm* - the full ISSC involves holding a flag at each hand but this is not used here). For example, in figure 4 the eight windows are a snapshot of behaviours which currently show movement towards letters E, N, L, K, I, E, T, and E respectively. There isn't any standardized set of movements used in movement imitation research and the reason for selecting this as a test set is that it is well defined, and it has the potential to support movement combinations of arbitrary length in a meaningful, principled way, since there is a correspondence to letters, thus a capacity to form syllables, words and so on.

Implementation platform

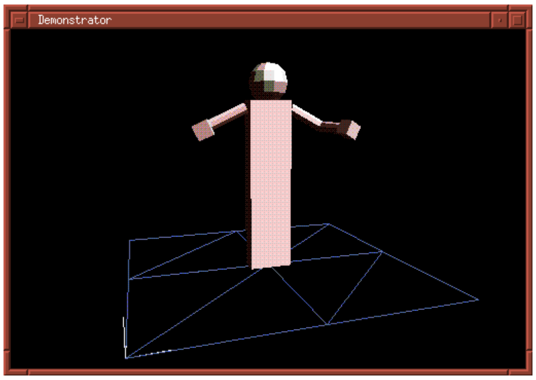

Figure 3: 'Bouncer', the thirteen DoF dynamically simulated robot

A thirteen degrees-of-freedom simulated robot was constructed and its dynamics simulated using the DynaMechs dynamical simulation library (McMillan et al, 1995). 'Bouncer' (figure 3) has three degrees of freedom at the neck joint, three at each shoulder, one at each elbow, and one at each wrist. Bouncer operates under the effect of gravity, so if no forces are applied to a joint, the connected body part moves toward the ground. The full experimental platform (figure 4) consists of two simulated robots, a demonstrator and an imitator, with similar body structure but with different dynamics (the imitator, in the results reported here, is 100% heavier that the demonstrator). The imitator is allowed to read the demonstrator's states (joint angles) in a crude simulation of visual capabilities. To account for the fact that in the case of the imitator, real vision and proprioception never results in perfectly correct values of the visually perceived demonstrator states and proprioceptively perceived imitator states, noise is added to them before they become available to the imitator.

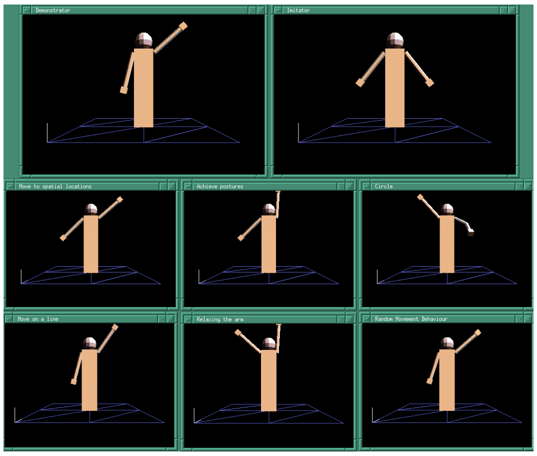

Figure 4: The full platform - demonstrator is top left, imitator top-right, and the six windows below display the behaviours that the imitator is currently generating internally

Implementation of the architecture

This section describes how the architecture was implemented for these experiments. This involves the description of the implementation of two architecture components: the behaviours and the forward models.

Implementation of the behaviours Behaviours were implemented as sets of key body postures, coupled with proportional-integral-derivative (PID) controllers (Astrom and Hagglund, 1996), that drive the controlled body part(s) through the key postures that constitute the behaviour. The PID controllers, one for each of the thirteen controlled joints, output torque values which are calculated by summing three terms: the proportional, integral, and derivative terms respectively, all of which are based on the error between the current state and the target state. If we take $e(t)$, the error at time t , to be the difference between the target state and the plant's state at time t, then the formula is as follows:

$$T(t) = K_p e(t) + K_i \int^t e(t)dt + K_d \frac{de(t)}{dt} \qquad (1)$$

where K_p, K_i, K_d are gains coefficients which were determined experimentally, and are allowed to adapt in order to cope with different movement speeds.

Implementation of the forward models The DynaMechs simulation package (McMillan et al, 1995) include libraries for simulating rigid body dynamics, and these were used in order to implement the forward models. The procedure involves four steps: (a) Applying the forces supplied by the inverse model, taking into account the current state (joint positions and velocities) of the robot. (b) Calculating all the forces exerted (including joint friction and gravity) and the AB inertias that are present in each joint. (c) Calculating the resulting accelerations recursively for each body part starting from the torso and moving towards the wrist. (d) Calculating the new state (joint positions and velocities).

Experimental results

This set of experiments consisted of the demonstrator performing a behaviour that was composed of a single action and the imitator observing it having a set of behaviours in its repertoire. The number of behaviours is not important since the behaviours run in parallel and they are independent of each other. Experiments with six behaviours are shown for graph clarity reasons.

Figure 5 shows an example confidence graph of an experiment where the demonstrator executes behaviour 'A', and the imitator has in its repertoire the behaviours 'A', 'B', 'C', 'D', 'E', and 'F'. All imitator's behaviours start initialy by having zero confidence, and end up having a confidence correlated with their similarity to the demonstrated behaviour. Since the demonstrator is demonstrating 'A', the imitator's 'A' behaviour gets the highest confidence, well above zero. All the other behaviours end up well below zero. It is interesting to note that while clearly all of them have not done well, some have done better than others. For example, behaviour 'B' which is more similar to 'A' than the others (its end position only differs from 'A' by one eighth of a circle on the right hand) has a higher confidence (although still negative) than 'E' and 'F'

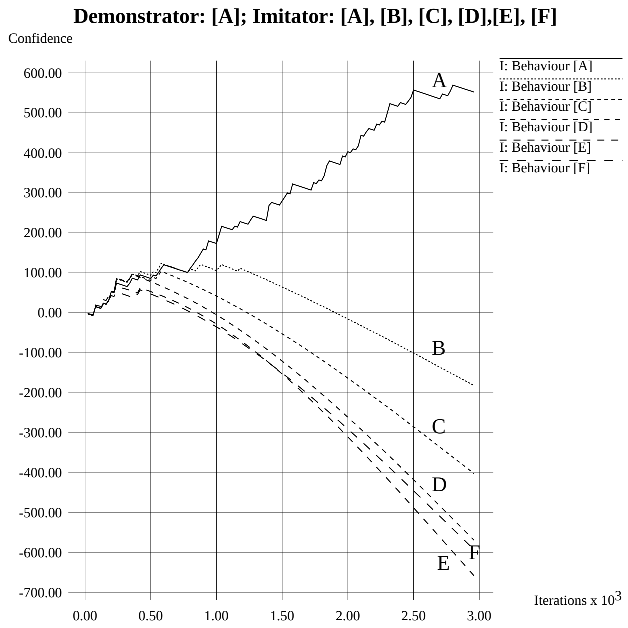

Figure 5: Confidences of imitator's internal behaviours A-F when demonstrator executes behaviour A;

for example, that differ from A with respect to both hands.

Similar results were obtained with a variety of different demonstrator and imitator dynamics, and with behaviours implementing sequences of letters (words).

Discussion

This architecture relies on, and highlights the importance of embodiment in cognition (Dautenhahn, 1997). Perception within this architecture is not an abstract process, classifying events as instances of generic classes, but utilises the specific generative models that the agent has, which have been developed as a results of having to control this specific body. Perception and motor action are tightly coupled: perception and recognition result from internally generating actions. There is a growing amount of evidence that indicates that perception, mental imagery, and motor action are tightly coupled in biological systems too. (Decety et al, 1994) have performed positron emission tomography experiments demonstrating that the same human brain regions are activated during passive observation of movements and imagining equivalent own movements. In addition, (Rizzolatti et al 1996) have discovered neurons in the monkey premotor area (brain area F5), which are firing during the production of a specific action, but also during the observation of this action performed by a demonstrator. Also, the use of prediction has important consequences in that it produces candidate actions before the observed action has been completed, and even if that action is never completed. (Meltzoff, 1996) has performed imitation experiments with human infants, demonstrating incomplete actions to them, which the infants have nevertheless correctly imitated. From an evolutionary perspective, use of prediction might have prevailed since it allows the animal to act/respond to the action of a conspecific before that (potentially non-beneficial to the observer) action has been completed.
Utilising computational and robotic architectures as models of real neural systems is a difficult task, but the biological data that are available indicate that the behaviour displayed by this architecture is towards the right direction. Current work is addressing the issue of learning the demonstrated behaviour when its not in the imitator's repertoire.

References

Astrom K.J. and Hagglund T., "PID Control", in *The Control Handbook*, Levine W.S. (ed.), IEEE Press, section 10.5, pp. 198-209, 1996.

Dautenhahn K., 1997. "I could be you - the phenomenological dimension of social understanding". In *Cybernetics and Systems Journal*, Special Issue on Epistemological Aspects of Embodied AI, 28(5), 417-453.

Decety J., Perani D. Jeannerod M., Bettinardi V., Tadary B., Woods R, Mazziota J.C, Fazio F., "Mapping motor representations with positron emission tomography", in *Nature*, pp. 600–602, Vol 371, 1994.

Demiris, J.; Rougeaux, S.; Hayes G.; Berthouze L.; Kuniyoshi, Y., 1997. Deferred Imitation of Human Head Movements by an Active Stereo Vision Head. In *Proc. of the 1997 IEEE Workshop on Robot-Human Communication*, Sendai, Japan, Sept. 1997.

Kaiser M. and Dillman R., "Building Elementary Robot Skills from Human Observation" in *Proc. of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996.

Kuniyoshi Y., Inaba M., and Inoue H., "Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance", in *IEEE Trans. on Robotics and Automation,* 10:6, 1994.

McMillan S, Orin D.E., McGhee R.B., "DynaMechs: An Object Oriented Software Package for Efficient Dynamic Simulation of Underwater Robotic Vehicles", in *Underwater Vehicles: Design and Control*, Yuh J. (ed.), TSI Press, chapter 3, pp. 73-98, 1995.

Meltzoff A., "The Human Infant as Imitative Generalist: A 20-Year Progress Report on Infant Imitation with Implications for Comparative Psychology", in *Social Learning in Animals: The Roots of Culture*, C. M. Heyes and B.G. Galef (eds), Academic Press.

Rizzolatti G.; Fadiga L.; Gallese V.; Fogassi L., 1996. "Premotor cortex and the recognition of motor actions". In *Cognitive Brain Research*, 3, 131-141.