Berthouze, L., Kaplan, F., Kozima, H., Yano, H., Konczak, J., Metta, G., Nadel, J., Sandini, G., Stojanov, G. and Balkenius, C. (Eds.)
Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems
Lund University Cognitive Studies, 123. ISBN 91-974741-4-2

From motor babbling to hierarchical learning by imitation: a robot developmental pathway

Yiannis Demiris and Anthony Dearden
Biologically-inspired Autonomous Robots Team (BioART)
Intelligent Systems and Networks Group
Department of Electrical and Electronic Engineering
Imperial College London, South Kensington Campus, SW7 2BT
{y.demiris, anthony.dearden}@imperial.ac.uk
http://www.iis.ee.ic.ac.uk/yiannis

Abstract

How does an individual use the knowledge acquired through self exploration as a manipulable model through which to understand others and benefit from their knowledge? How can developmental and social learning be combined for their mutual benefit? In this paper we review a hierarchical architecture (HAMMER) which allows a principled way for combining knowledge through exploration and knowledge from others, through the creation and use of multiple inverse and forward models. We describe how Bayesian Belief Networks can be used to learn the association between a robot's motor commands and sensory consequences (forward models), and how the inverse association can be used for imitation. Inverse models created through self exploration, as well as those from observing others can coexist and compete in a principled unified framework, that utilises the simulation theory of mind approach to mentally rehearse and understand the actions of others.

1. Introduction

Equiping a robot with the ability to learn enables it to adapt its behaviour in order to improve its performance on certain tasks. Similarly, imitation (among other social learning algorithms) has been shown to be a powerful way for transferring behaviour patterns from one agent to another in order to reduce the time it takes the observer to achieve a task. Although powerful frameworks exist for both social and a-social learning, there hasn't been a lot of work in combining techniques from the two domains in a unified principled representational frameworks. Such frameworks would allow a robot to combine (and use) knowledge acquired from self-exploration, with knowledge acquired through observation, and allow both to be used in a unified way.

In this paper we review our architecture, HAMMER (Hierarchical, Attentive, Multiple Models for Execution and Recognition), which adopts the simulation theory of mind approach (Carruthers and Smith, 1996) to understanding the actions of others, using a distributed network of coupled inverse (Narendra and Balakrishnan, 1997) and forward (Miall and Wolpert, 1996) models. New models can be added either through self-exploration (Dearden and Demiris, 2005), or by observing others, through imitation (Demiris and Hayes, 2002, Demiris and Johnson, 2003).

First we describe how Bayesian Belief Networks can be used to learn the mapping between motor commands and corresponding visual feedback states (the forward models). We subsequently invert this mapping to learn basic inverse models. We then proceed to explain how having the same structure, a coupled inverse and forward model allows the robot to either execute, rehearse or perceive an action. Having a distributed network of coupled inverse and forward models helps us develop hierarchies of increasingly complex inverse models, and combine both learning through self exploration and learning from others. We conclude by outlining current challenges in this developmental framework, namely object-oriented actions, and development of more complex body schemas.

2. Background

Current approaches in robotics emphasize the need for mechanisms to allow a robot to adapt to new situations and be able to perform new tasks. Environmental conditions, task requirements and context cannot be assumed to be known in advance. Robot Learning mechanisms have thus received extensive attention in the last few years (Conell and Mahadevan, 1993, Franklin et al., 1996, Demiris and Birk, 2000, Schaal, 2002). These techniques can be generally grouped in social and asocial learning. In the latter, the robot attempts to learn to perform a task by interacting with its environment, without considering the solutions that other agents in its environment possesss with respect to this task. Techniques can be generally divided into supervised and unsupervised learning, with one of the most widely used being reinforcement learning (Sutton and Barto, 1998).

On the other hand, recent approaches have emphasized the importance of social learning in robotics, including learning by observation, demonstration and imitation. Equipping robots with the ability to imitate enables them to learn to perform tasks by observing a human demonstrator (Schaal, 1999). In the center of this ability lies a mechanism that matches demonstrated actions with motor actions available to the robot (Demiris and Hayes, 2002, Billard, 2000, Schaal et al., 2003). Several architectures have been proposed for implementing this mechanism (for reviews see (Breazeal and Scassellati, 2002, Schaal, 1999, Schaal et al., 2003, Dautenhahn and Nehaniv, 2002)), including a number of proposals utilizing a shared substrate between execution, planning, and recognition of actions (Demiris and Hayes, 2002, Schaal et al., 2003, Wolpert et al., 2003). This methodological shift, compared to other successful approaches to learning by demonstration (Kuniyoshi et al., 1994) was inspired by the discovery of the mirror system (di Pellegrino et al., 1992, Decety et al., 1997), which indicated that, at least in primates, there is indeed a shared neural substrate between the mechanisms of action execution and those of action recognition. Apart from being compatible with the motor theories of perception, from an engineering perspective this approach is also attractive since it allows reuse of subsystems in multiple roles.

Despite extensive work on social and asocial learning, there isn't a lot of work being done in bringing these together. In this paper we will review our computational framework, HAMMER, that allows a principled way for combining knowledge from the two sources, and use them to incrementally learn more complex structures. We will start by explaining how our system can learn primitive forward and inverse models by motor babbling, and demonstrating this on an ActivMedia Peoplebot, learning to predict the consequences of its motor commands on its grippers. We briefly demonstrate how these primitive forward models can be inverted and used to imitate gestures done by a demonstrator. Having primitive inverse and forward models, we explain how they can be used to build more complex hierarchical inverse models (Demiris and Johnson, 2003).

3. Learning through self-exploration

Drawing inspiration from motor babbling in infants (Meltzoff and Moore, 1997), we aim at building a system that enables a robot to autonomously learn a forward model with no a priori knowledge of its motor system or the external environment. The robot sends out random motor commands to its motor system and uses this, together with the information from its vision system to learn the structure and parameters of a Bayesian belief network (BBN), which represents the forward model. This autonomously learnt model can then be used to enable the robot to predict the effects of its own actions, or imitate the actions of others.

3.1 Learning forward models

Bayesian Belief Networks (BBN) are well suited for representing forward models which can be learnt. They enable the causal relationship between the robot's motor commands, its motor system and the observations from the robot's sensor system to be modelled and learnt. Each motor command, state of the robot or sensor observation is represented as a random variable in the BBN, and the causal relationships between them are represented with arcs. The BBN therefore represents a learnt probability distribution across the previous N motor commands, $M_{1:N}[t-d]$, the P current states $S_{1:P}[t]$, and the P observations of the state, $O_{1:P}[t]$. The variable d represents the delay between a motor command being issued and robot's state changing; in real robotic systems it cannot be assumed that the effect of a motor command will occur after just one time-step, so this is a parameter that must be modelled and learnt. Forward models predict the consequence of a motor command. Therefore, the BBN can be used as a forward model by performing inference with the network to calculate the probabilities of robot states or observations given a particular set of motor commands: P(S[*t*]|M[*t*-d]=*m)* or P(O[*t*]|M[*t*-d]=*m)*. Figure 1 shows this prototype network structure.

Several aspects of this BBN structure need to be learnt. The robot first needs to learn the number of state variables, P, and what they represent. Here, this is done by using information suppled from a the computer vision system, which is able to track objects in a scene using no prior information about the number or properties of the objects. Objects are found by clustering regions in the image with similar position and movement properties. The state of each object found can be represented with a node on the BBN, and the tracking information from the vision system can be represented with an associated observation node. The remaining parts of the structure that need to be learnt are which motor commands control which state variables, and which observations

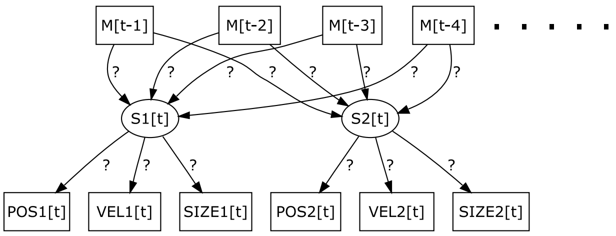

Figure 1: *The BBN of a forward model. The underlying state of the robot is represented with hidden nodes. The question marks represent the part of the structure which needs to be learnt: the causal association between the motor commands and the state of the robot, and the state of the robot with the observations from the computer vision system.*

from the computer vision system are the correct observation of an object's state.

To train a BBN, a set of evidence is required: a data set containing actual observed values of the observed nodes in the network. This is a set of motor commands executed at each time step, $m_{1:N}[t-d]$, together with the observations of the state $o_{1:P}[t]$. Two issues need to be addressed: how to create this set of evidence, and how to use this evidence to learn the structure and parameters of the BBN. The inspiration for solving these problem here is taken from developmental psychology. (Gopnik and Schulz, 2004) compares the mechanism an infant uses to learn to that of a scientist: scientists form theories, make observations, perform experiments, and revise existing theories. Applying this idea here, forward models can be seen as a theory of the robot's own world and how it can interact with it. The process a robot uses to obtain its data and infer a forward model from it is its experiment. To perform an experiment, a robot decides what motor commands to use, gathers a corresponding set of evidence and uses this evidence to learn the BBN structure. The motor commands are chosen at random using a Markov model to randomly change the current motor command. To learn the structure and parameter of a BBN, it is necessary to perform a search through the space of possible structures, with the goal of finding the one that maximises the log-likelihood of the model given the data. This search space would normally be extremely large. However, for a forward model the number of possible structures is made much smaller because of the constraints placed on the dependence of particular nodes, e.g. the motor command nodes can only be connected to the robot's state nodes.

3.2 Experiments

An experiment was carried out to show how forward models can be autonomously learnt and used. The task was to learn the forward model for the movement of the robot's grippers; the robot needed to learn to predict the effects of the motor commands. No prior information about the appearance of the grippers or the nature of the motor commands was specified. The robot was to perform an experiment: it babbled with its motor commands, gathered evidence of the motor commands and corresponding observations, and then learnt the relationship between these using a BBN.

When the gripper motor system was babbled, the vision system correctly calculated and tracked the positions of the moving grippers in the scene. This provided the random variables for the states $\mathbf{S}[t]$ and observations $\mathbf{O}[t]$ in the BBN for the forward model: each gripper's state was represented as a discrete node for the state, and a set of continuous nodes for the observation, both of which were represented as a Gaussian distribution. The observations, $\mathbf{O}[t]$ could be either the velocity VEL[t] or the position POS[t] of the tracked grippers. Learning the structure of the BBN of the forward was done by searching through the space of structures for different motor delays and observations to find the structure that maximised the likelihood of the model given the data. This was performed by simutaneously training each possible forward model structure, and tracking its log-likelihood given the data. The parameters of each BBN were learnt online using the recursice expectation maximisation (EM) algorithm. Learning multiple forward models simultaneously allows the best one at any particular step of the training process to be immediately available. As shown in figure 2 the motor command was learnt to be M[t -11], corresponding to a delay of 200ms, and the best observation was correctly learnt to be VEL[t].

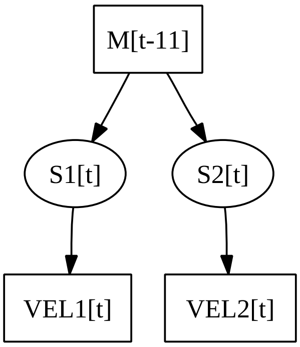

Figure 2: *The BBN of the forward model which was learnt by the robot*

Once the BBN was learnt, it could be used as a for-

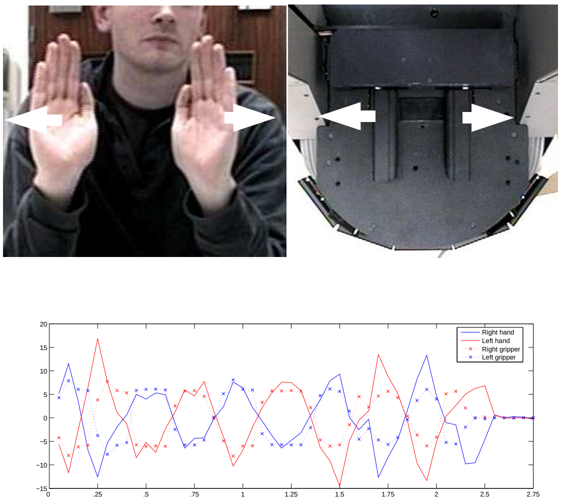

Figure 3: *Imitation using a forward model. The top images are corresponding frames from the human demonstration (left) and the robot's imitating grippers (right). The graph shows the trajectory of the demonstrating hands and the imitating trajectory of the grippers.*

ward model: a prediction for a given motor command is the probability of an observation of a gripper moving at a particular velocity given a particular motor command: $\mathrm{P}(\mathrm{VEL}[t]|\mathrm{M}[t\text{-}11]=m)$. Alternatively, the most likely gripper command to achieve a particular movement can be inferred: $\mathrm{P}(\mathrm{M}[t\text{-}11]|\mathrm{VEL}[\mathrm{t}]=\mathrm{v})$. This can now be used a primitive inverse model - it gives a conditional probability distribution of the motors commands given a particular observation, so taking the most likely motor command, the robot can try and recreate a particular movement observation. This can be used for imitation:. if the observations given to the inverse model are of a human's hands, the robot is able to reproduce the motor commands that would be most likely to recreate that human movement in its own gripper system. Figure 3 shows the results of a simple imitation experiment: the robot is able to imitate a simple hand waving motion. The human's hand movements are tracked and recognised using the same vision system. The observations are used as evidence for the BBN already learnt by the robot. This network is then used to predict the most likely motor command that would produce this observation. These motor commands are given to the robot enabling it to perform the action that best replicates the observed movement. Other forward and inverse models can be created and used this way (but see the concluding discussion about limitations of the vision-based comparison process) for the other possible degrees of freedom of the robot.

4. Learning from others

Having established this critical correspondence between learning forward models and their use as inverse models through inverse mapping, we are now in position to incorporate into the same architecture the ability to learn from others. We do so by utilising the simulation theory of mind approach (Carruthers and Smith, 1996), which postulates that when we observe others, we put ourselves in their position, and internally simulate their actions to understand what they are doing. In (Demiris and Johnson, 2003) we have done so using HAMMER, a competitive, multiple inverse and forward models architecture.

4.1 Coupling Inverse and forward models

The fundamental structure of HAMMER is an inverse model paired with a forward model (figure 4). In order to execute an action within this structure, the inverse model module receives information about the current state (and, optionally, about the target goal(s)), and it outputs the motor commands that it believes are necessary to achieve or maintain the implicit or explicit target goal(s). The forward model provides an estimate of the next state. This is fed back to the inverse model, allowing it to adjust any parameters of the behaviour (an example of this would be achieving different movement speeds (Demiris and Hayes, 2002)). Additionally, the same structure can be used in order to match a visually perceived demonstrated action with the imitator's equivalent one. This is done by feeding the demonstrator's current state as perceived by the imitator to the inverse model and having it generate the motor commands that it would output if it was in that state and wanted to execute this particular action. 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 demonstrator's actual state at the next time step. This comparison results in an error signal that 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.

HAMMER consists of several of the structures described above, operating in parallel (Demiris and Hayes, 2002, Demiris and Johnson, 2003). Fig. 5 shows the basic structure. When the demonstrator executes an action the perceived states are fed into the imitator's available inverse model. This generates motor commands that are sent to the forward models. The forward models generate predictions about the demonstrator's next state: these are compared with

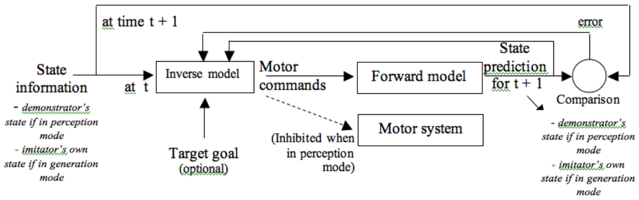

Figure 4: *Coupling an inverse with a forward model; this structure can be used to execute an action, as well as rehearse it and match it with an external observation (Demiris and Hayes, 2002).*

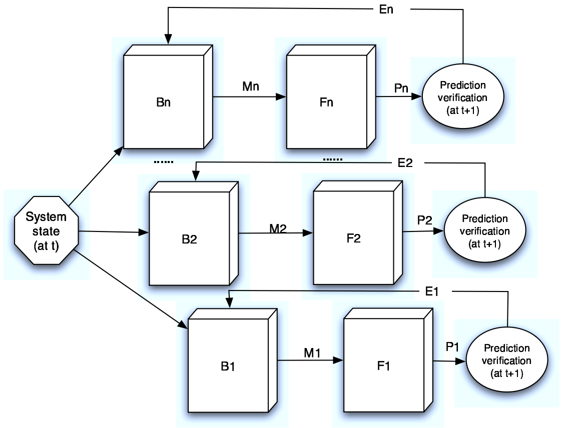

Figure 5: *The basic architecture, showing multiple inverse models (B1 to Bn) receiving the system state, suggesting motor commands (M1 to Mn), with which the corresponding forward models (F1 to Fn) form predictions regarding the system's next state (P1 to Pn); these predictions are verified at the next time state, resulting in set of error signals (E1 to En)*

the actual demonstrator's state at the next time step, and the error signal resulting from this comparison affects the confidence values of the inverse models. At the end of the demonstration (or earlier if required) the inverse model with the highest confidence value, i.e. the one that is the closest match to the demonstrator is selected. This architecture has been implemented in real-dynamics robot simulations and real robots (Demiris and Johnson, 2003, Johnson and Demiris, 2004) and has offered plausible explanations and testable predictions regarding the behaviour of biological imitation mechanisms in humans and monkeys (review in (Demiris and Johnson, 2005)).

4.2 Learning hierarchical structures

More recently we have designed and implemented a hierarchical extension (Demiris and Johnson, 2003), to this arrangement: primitive inverse models are combined to form higher more complex sequences, with the eventual goal of achieving increasingly more abstract inverse models. Given a set of primitive inverse models and forward models (for example in (Demiris and Johnson, 2003) we used raise and lower gripper platform and open and close gripper) more complex ones can be created through observation (details of the learning algorithm in (Demiris and Johnson, 2003)). Incremental learning and scaffolding have already been highlighted as crucial factors in development (Lungarella et al., 2003, Weng, 2004).

One of the crucial issues in the learning of more complex structures is that of resetting (Demiris, 2002). If the demonstrator is executing a primitive inverse model, then selection of the appropriate inverse model for the imitator is performed by a winner-take-all approach: the inverse model with the higher confidence is selected. However, in cases where the demonstrator performs a more complex sequence, the sychronization and timing of the activation of the inverse models in the imitator's repertoire become critical.

More specifically, the fact that an inverse model already present in the imitator's repertoire might appear *at any point during the demonstration* as part of the demonstrated sequence requires the existence of a mechanism that can reinitialise (reset) the imitator's inverse model during the demonstration. That will involve re-running the initialisation steps that each inverse model takes at the start of the experiment, i.e. the state of the inverse model is set to that of the demonstrator, its confidence is set to zero, and the inverse model starts moving towards its first (or only) goal, so even if the inverse model has a low confidence because it didn't match well previous parts of the demonstrated sequence, it will be given a new chance for the current part.

The crucial issue here is, during the demonstration, when should an inverse model in the imitator's repertoire be reset? The available options are: (a) Reset upon completion of this inverse model. (b) Reset when the confidence level of this inverse model has dropped below a certain value (c) Reset when all inverse model in the imitator's repertoire have been completed

We have performed experiments (Demiris, 2002) investigating the behavioural effects of these three options, as well as combinations; the best results were obtained through the adoption of the third option, where inverse models are reset when all inverse models have been completed. This was implemented through a mutual-inhibition mechanism: each inverse model inhibits the resetting of all other inverse models until it reaches its goal. When all inverse models have reached their goals, all inhibitions have been removed, and all inverse models are reset. This has the side effect that we might get better performance if we keep the different hierarchical inverse models balanced in terms of length; we are currently investigating whether this is indeed the case.

5. Discussion

5.1 Advantages and disadvantages from a distributed approach

HAMMER stores procedural knowledge in the form of multiple inverse models. By doing so, it allows different solutions to a problem to co-exist without considerable interference between them. This is a significant advantage over approches where a global centralised controller is learned, and catastrophic interference (Schaal and Atkeson, 1998) is observed. This is particularly true for situations where the learning data are obtained incrementally, rather than being available in advance (Schaal, 2002).

However our architecture will over time continue to learn an increasingly high number of inverse models. Although the inverse models run in parallel, they do compete on the basis of the quality of predictions they generate, and as discussed in the previous section, when placed in sequences and hierarchical structures, can mutually inhibit each other, thus the demand for an efficient organisation. One way of dealing with the combinatorial explosion is to remove or combine inverse models based for example on usage, and thus implementing forgetting or memory consolidation mechanisms (Robertson et al., 2004).

5.2 Open challenges

There are a number of challeges that remain unsolved:

- In our current experiments, forward models are learned between the motor commands and the image plane based visual perception of the effects of these actions. A generalisation of these to more abstract representations (for example a 3-D body schema) will allow more complex relationships to be addressed (the visual information contained in a Japanese bow for example, is different when you perform it from when you observe it)

- Object oriented actions - although we have performed a number of experiments learning hierarchical structure involving object oriented actions (Johnson and Demiris, 2004), we have used hardwired object oriented action primitives. We are in the process of applying the BBN approach above to learning the effects of motor commands on objects.

Elsewhere (Demiris and Khadhouri, 2005) we also elaborate on the issue of top-down control and joint attention (Kozima et al., 2003, Nagai et al., 2003) in the selection of features to observe and imitate. Despite the need for further research on these highlighted challeges, our approach so far demonstrates a developmental pathway that allows a robot to combine learning from self-exploration and learning from imitation in a principled way.

Acknowledgements

This research was supported by the Engineering and Physical Sciences Research Council (EPSRC - Grant GR/S11305/01 to YD, and a Doctoral Training Award to AD) and the Royal Society. Thanks to all the BioART members, including Matthew Johnson, Bassam Khadhouri, Gavin Simmons and Paschalis Veskos, to Oliver Grau of BBC Research and Development, and to three anonymous reviewers for their feedback.

References

Billard, A. (2000). Learning motor skills by imitation: a biologically inspired robotic model. *Cybernetics and Systems*, 32:155–193.

Breazeal, C. and Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11):481–487.

Carruthers, P. and Smith, P. K. (1996). *Theories of Theories of Mind*. Cambridge University Press.

Conell, J. H. and Mahadevan, S., (Eds.) (1993). *Robot Learning*. Kluwer Academic Publishers.

Dautenhahn, K. and Nehaniv, C. (2002). *Imitation in Animals and Artifacts*. MIT Press, Cambridge, MA, USA.

Dearden, A. and Demiris, Y. (2005). Learning forward models for robotics. In *Proceedings of IJCAI*.

Decety, J., Grezes, J., Costes, N., Perani, D., Jeannerod, M., Procyk, E., Grassi, F., and Fazio, F. (1997). Brain activity during observation of actions: Influence of action content and subject's strategy. *Brain*, 120:1763–1777.

Demiris, Y. (2002). Imitation, mirror neurons, and the learning of movement sequences. In *Proceedings of the International Conference on Neural Information Processing (ICONIP-2002)*, pages 111–115. IEEE Press.

Demiris, Y. and Birk, A., (Eds.) (2000). *Interdisciplinary approaches to robot learning*. World Scientific.

Demiris, Y. and Hayes, G. (2002). Imitation as a dual route process featuring predictive and learning components: a biologically-plausible computational model. In Dautenhahn, K. and Nehaniv, C., (Eds.), *Imitation in Animals and Artifacts*. MIT Press.

Demiris, Y. and Johnson, M. (2003). Distributed, prediction perception of actions: a biologically inspired architecture for imitation and learning. *Connection Science*, 15(4):231–243.

Demiris, Y. and Johnson, M. (2005). Simulation theory of understanding others: A robotics perspective. In Dautenhahn, K. and Nehaniv, C., (Eds.), *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press.

Demiris, Y. and Khadhouri, B. (2005). Hierarchical, attentive multiple models for execution and recognition. In *Proceedings of the IEEE ICRA Workshop on Social Mechanisms of Robot Programming by Demonstration.*

di Pellegrino, G., Fadiga, L., Fogassi, L., Gallese, V., and Rizzolatti, G. (1992). Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91:176–180.

Franklin, J., Mitchell, T., and Thrun, S., (Eds.) (1996). *Recent advances in Robot Learning.* Kluwer Academic Publishers, Boston, MA, USA.

Gopnik, A. and Schulz, L. (2004). Mechanisms of theory formation in young children. *Trends in Cognitive Sciences*, 8(8):371–377.

Johnson, M. and Demiris, Y. (2004). Abstraction in recognition to solve the correspondence problem for robot imitation. In *Proceedings of TAROS*, pages 63–70, Essex.

Kozima, H., Nakagawa, C., and Yano, H. (2003). Attention coupling as a prerequisite for social interaction. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 109–114. IEEE Press.

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(4):151–190.

Meltzoff, A. N. and Moore, M. K. (1997). Explaining facial imitation: a theoretical model. *Early Development and parenting*, 6(2):157.1–14.

Miall, R. C. and Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, 9(8):1265–1279.

Nagai, Y., Hosoda, K., Morita, A., and Asada, M. (2003). A constructive model for the development of joint attention. *Connection Science*, 15(4):211–229.

Narendra, K. S. and Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2):171–187.

Robertson, E. M., Pascual-Leone, A., and Miall, R. C. (2004). Current concepts in procedural consolidation. *Nature Reviews Neuroscience*, 5.

Schaal, S. (1999). Is imitation learning the way to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

Schaal, S. (2002). Learning robot control. In Arbib, M., (Ed.), *The Handbook of brain theory and neural networks*, pages 983–987. MIT Press.

Schaal, S. and Atkeson, C. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084.

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Phil. Trans. R. Soc London B*, 358:537–547.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: an introduction.* MIT Press.

Weng, J. (2004). Developmental robotics: theory and experiments. *Journal of Humanoid Robotics*, 1(2):199–236.

Wolpert, D. M., Doya, K., and Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Phil. Trans. of the Royal Society of London B*, 358:593–602.