Learning Forward Models for Robots

Anthony Dearden, Yiannis Demiris
Department of Electrical and Electronic Engineering
Imperial College London
Exhibition Road, London, SW7 2BT
E-mail: {anthony.dearden, y.demiris}@imperial.ac.uk

Abstract

Forward models enable a robot to predict the effects of its actions on its own motor system and its environment. This is a vital aspect of intelligent behaviour, as the robot can use predictions to decide the best set of actions to achieve a goal. The ability to *learn* forward models enables robots to be more adaptable and autonomous; this paper describes a system whereby they can be learnt and represented as a Bayesian network. The robot's motor system is controlled and explored using 'motor babbling'. Feedback about its motor system comes from computer vision techniques requiring no prior information to perform tracking. The learnt forward model can be used by the robot to imitate human movement.

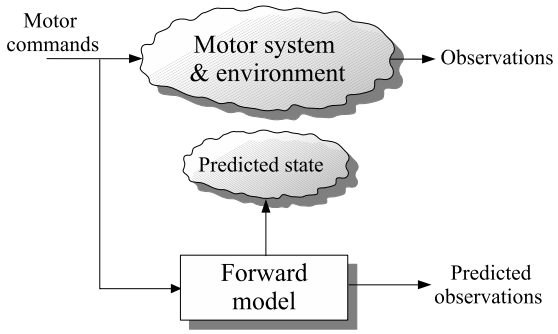

Figure 1: *A forward model*

1 Introduction

The purpose of any robot is to interact with its environment. There is, however a separation between the effects a robot can produce in the environment and its mechanisms for producing them, its motor commands. To overcome this, inverse and forward models can be used [Jordan and Rumelhart, 1992]. In robotics, forward models enable a robot to predict the consequences of its actions: given the current state of the robot they predict how a set of motor commands will influence the state of the robot or its environment, or how the observations will be perceived by the robot's sensors, as shown in figure 1. Conversely, inverse models are used to output the necessary motor commands to achieve or maintain a desired goal. Both these *internal models* have been hypothethised to be used in human motor control [Wolpert *et al.*, 1998]. The ability to predict the consequence of actions has several uses in robotics, such as allowing mental rehearsal of possible actions or imitating the actions of humans or other robots [Demiris, 2002].

Practically any environment a robot works in will change, or have properties which cannot be modelled beforehand. Even if the environment is assumed to be completely predictable, endowing the robot with this knowledge may be beyond the abilities or desires of its programmer. A truly autonomous robot, therefore, needs to be able to learn and adapt its own forward models. Forward models of dynamic and uncertain environments, and how they can be learnt, is the topic which is investigated in this paper. Although forward models are used in a different context, there are similarities here to work on calibrating action-sensor models in robotics e.g. [Roy and Thrun, 1999]. A robot learning about itself and its environment is in the position of being an active part of the system it is trying to learn about; this situation draws interesting parallels with learning in human infants, with mechanisms such as babbling (issuing random motor commands) used by infants to learn how to control their own motor system [Meltzoff and Moore, 1997].

A system is presented here that enables a robot to autonomously learn a forward model with no prior knowledge of its motor system or the external environment. Information about the effects of the robot's actions are captured via a vision system that clusters low-level image features to automatically find and track moving objects in a scene. The robot sends out random motor commands to its motor system and receives information back from the vision system. This set of evidence is used to learn the structure and parameters of a Bayesian network, which represents the forward model. This model can then be used to enable the robot to predict the effects of its actions. By observing the gestures of a human with its own vision system, the robot can imitate the human gesture using the forward model it has learnt of its own motor system.

in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1440-1445, Edinburgh, 2005.

2 Representing forward models with Bayesian networks

Bayesian networks [Pearl, 1988] are an ideal method for representing forward models that can be learnt. They offer a way of representing the causal nature of a robot's control system within a rigorous probabilistic framework. The system here is aimed at learning and representing the causal associations between the robot's motor commands, the robot's state, and the observations of the robot's state that are received from the vision system. The motor commands and state of the robot are represented as random variables in the Bayesian network, and the causal relationships between them are shown with arcs. The Bayesian network 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. Figure 2 shows this structure.

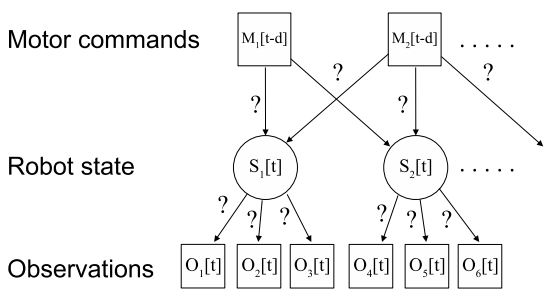

Figure 2: *The 'template' Bayesian network of a forward model. The question marks represent the part of the structure which needs to be learnt: the causal association between the motor commands and the state of the robot, and which observations are associated with each object in the scene. Rectangular nodes correspond to observed nodes.*

The N motor random variables represent specific low-level motor commands of the robot. They could be discrete, e.g. 'open' or 'close' robot's gripper, or continuous, e.g. to set the height of the gripper. When the robot is issuing these commands there is no uncertainty as to their value, so they can be modelled as observed nodes. Alternatively, when the robot is observing the actions of another robot, or a human, whilst using its own forward model, they will be hidden nodes because the Bayesian network will receive no direct evidence of their value.

The P observations of the state-space $O[t]$ are represented with square nodes descended from $S[t]$ on figure 2. The model treats $O[t]$ as noisy observations of the actual state space so that noise from the vision system and other uncertainty can be modelled. Evidence about the actual state of the robot therefore enters the system here. How the state space, $S[t]$, of the robot will be represented depends on the information received from the computer vision system. The state variables are all hidden nodes, because the robot's knowledge of itself only comes from noisy observations of the state, $O[t]$.

3 How a robot can learn and use forward models

The structure for a forward model described in section 2 require several aspects to be learnt. Firstly, a representation of the state of the robot or the environment is needed. The richest source of this information comes from the robot's own vision system. However, this is also the most challenging source: information comes in the low-level format of pixels. Furthermore, if the robot is seeking to learn its forward models it has no prior information about what to track, because the environment is unknown. To extract information from a scene, a method is needed of automatically initialising and tracking the objects in it. The system presented here works by clustering tracked low-level features in the image according to their position and dynamics. This allows the position of objects in the scene to be tracked without any knowledge of the objects having to be supplied.

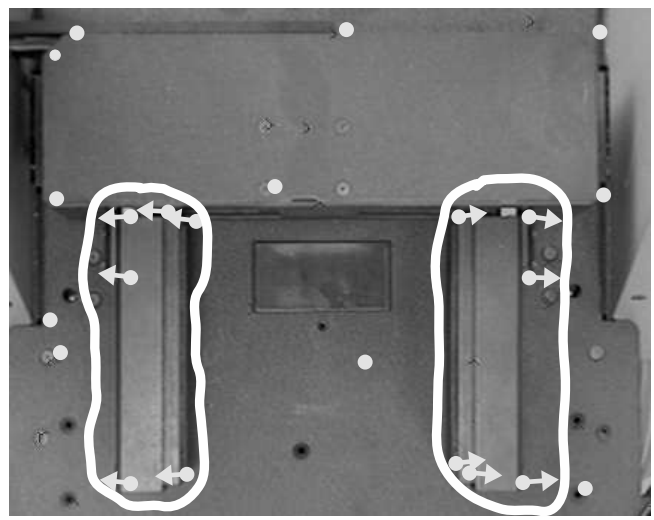

Figure 3: *The vision system finds moving objects in the scene by clustering together optical flow points.*

The low-level point features used are tracked using the Lucas-Kanade optical flow algorithm [Lucas and Kanade, 1981]. The tracked points are clustered using the k-means algorithm according to their position and velocity. One drawback of this method of clustering is that the number of clusters must be known beforehand. This is overcome by first excluding stationary points and extracting increasing numbers of clusters. The error of a particular number of clusters is then taken to be the average squared distance of the optical flow points from the centre of the cluster they are part of. The cluster number used is the one at which the curvature of this error with respect to the number of clusters is minimised [Hoppner *et al.*, 1999]. The positions, velocities and sizes of each of these clusters can now be used as the observations, O[t], of objects for the forward model learning system. The

in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1440-1445, Edinburgh, 2005.

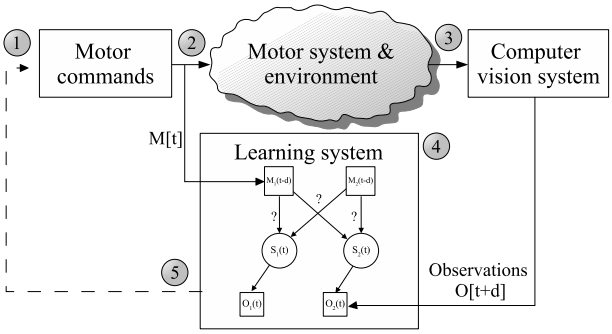

Figure 4: *How the robot can learn about its environment. The stages 1-5 involved in the experiment are outlined in the text.*

actual state of the motor system, S[*t*], is taken to be a hidden discrete node for each object in the scene found by the vision system, and this will be connected to a continuous observation.

With all the possible nodes in the Bayesian network specified, the final task is to learn how the motor commands interact with the state, and train the parameters of the Bayesian network so that its predictions are as accurate as possible. To train a Bayesian network, 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 *N* motor commands executed at each time step, $M_{1:N}[t-d]$, together with the *P* observations of the state $O_{1:P}[t]$, the size, position or velocity of an object. The problem now is how to use this data to learn the structure and parameters of the Bayesian network. An inspiration for solving the problem of how to learn this Bayesian network can be taken from developmental psychology. Gopnik and Meltzoff compare the mechanism an infant uses to learn to that of a scientist: scientists form theories, make observations, perform experiments, make generalisations and revise existing theories [Gopnik and Meltzoff, 1998]. The notion of forming theories about the world is already dealt with here: a forward model can be seen as a theory of the robot's own world. The process a robot uses to obtain its data and infer a forward model from it is its *experiment*. An outline of what a robot's experiment involves is as follows, and is highlighted in figure 4:

1. A particular motor command, $M_1[t]$ is chosen;
2. A value for this motor command is sent to the robot's motor system;
3. The vision system returns a set of observations of the scene, $O_{1:P}[t+d]$, for example, moving forward $v\ ms^{-1}$ in direction *x;*
4. This training set of data, $M_1[t]$ and $O_{1:P}[t+d]$, is used to train the Bayesian network to develop a forward model;
5. The whole process is repeated, either with a different value for the current motor command, or a different motor command altogether.

The open issues here relate to the last two stages. Firstly, if the structure of the Bayesian network is not completely known, how should the data be used to learn the structure and parameters of the network? Most methods for learning structure from data are based on performing 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 [Friedman, 1997; Neapolitan, 2004]; the higher the log likelihood, the more accurate a particular network is at predicting the data. This is subject to a constraint on the complexity of the model, as the structure with the highest likelihood will be the one with connections between every node. In this paper the network structure is learnt as part of the experiment by performing a search through the set of possible structures and choosing the one which maximises the log-likelihood. This search is performed online by training and evaluating simultaneously a set of candidate models. The search space is the set of models with different delays in motor commands and different observation nodes for each possible object.

The second issue is how particular motor commands and their values should be chosen so as to learn the most accurate forward model. Unlike most machine learning situations, with a forward model the robot is an active part of the Bayesian network structure it is attempting to learn. This situation, referred to as *active learning*, has received relatively little attention in Bayesian networks literature. Active learning for Bayesian networks has been discussed for unknown parameters and unknown structure [Tong and Koller, 2000; 2001]. The method of choosing motor commands in the experiment in this paper was inspired by the idea of babbling in infant motor skill learning [Meltzoff and Moore, 1997]; the motor commands were chosen from a random walk through a Markov model.

4 Experiments to learn a forward model

An experiment was carried out to show how forward models can be autonomously learnt and used. The robot used was an ActivMedia PeopleBot, as shown in figure 5. 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, observed what happened, and then learnt the relationship between the motor command and the observation using a Bayesian network; this process was repeated until an accurate model had been learnt.

The motor command in this case was limited to just one degree of freedom: the robot's grippers, which could be either opened, closed or halted. The motor commands were decided at random using a random walk through the Markov model shown in figure 6. Motor commands were sent according to the current state of the Markov model, and the state would change each time-step according to the transition probabilities shown on the figure. The parameters of the Markov model were chosen so that the motor system stays in one babbling state just long enough to get enough information to train its forward model before moving to the next state.

When the motor commands were sent, the grippers moved; the vision system correctly calculated and tracked the posi-

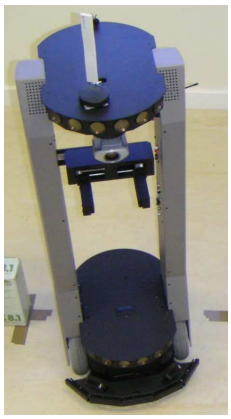

Figure 5: *An ActivMedia PeopleBot.*

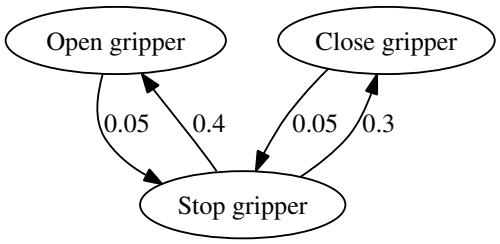

Figure 6: *Markov model of gripper babbling. The values on the arcs represent the transition probabilities of going from one state to the next; self transitions are not shown.*

tions of the moving grippers in the scene: figure 7 shows the two grippers being located and tracked. The tracking system worked well in this situation: black objects are automatically being tracked even against the black background of the robot's base.

The vision system provided the random variables for the states $\boldsymbol{S}[t]$ and observations $\boldsymbol{O}[t]$ in the Bayesian network for the forward model. In this situation the two objects, the grippers, were automatically added to the Bayesian network. Each gripper 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 set of possible observations, $\boldsymbol{O}[t]$, was the velocity, VEL[t], the position, POS[t], or size, SIZE[t] of the tracked objects. The basic template for the structure of the forward models to be learnt is shown by the Bayesian network in figure 8.

Even though the forward model of the robot's gripper was relatively simple, this example highlighted how much the robot needed to learn: all the parameters and two important properties of the structure needed to be learnt. Firstly, it is not known how long the delay is between the motor command being issued and the gripper changing state. This is shown in figure 8 as the multiple possible motor commands connected to each state; only the four previous commands are shown to keep the diagram simple. Secondly, it was not known which observation from the vision system is the best

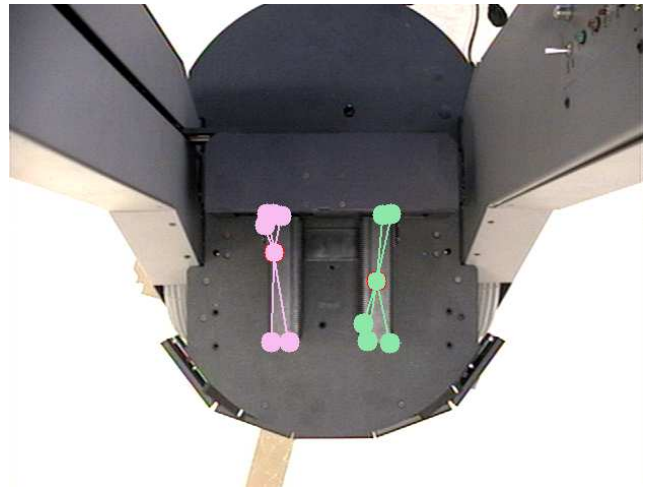

Figure 7: *The tracking system working on the grippers. The system has correctly identified two moving objects, which are identified as grouped primitives shown with the same colour.*

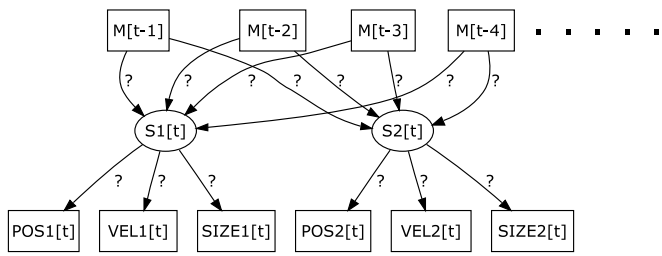

Figure 8: *The template Bayesian network for the gripper forward model. The vision system has supplied the information that two objects can be interacted with: the two grippers. The robot still has to learn what the delay,* d*, is from action to effect and which of the observations the change of state of each object represents, a translation or a change in velocity, and all the parameters that specify the Bayesian network.*

one to use as part of the forward model: the velocity, position or size. The task of the structure learning algorithm here is therefore to maximise the log-likelihood of the data given the model, which in this case is $\log\left(P\left(M, O \mid G\right)\right)$, where M is the set of motor commands over the experiment, O the set of corresponding observations and G is the directed acyclic graph representing the structure of the model, together with the parameters of the model. This value is calculated as part of the parameter learning process, as a particular model is trained to the data.

For each model, the parameters of the Bayesian network were learnt using the expectation maximisation (EM) algorithm, with the inference stage being performed with the junction-tree algorithm [Pearl, 1988]. In this experiment, the search space is small enough for the robot to consider each possible candidate structure simultaneously. If the previous 20 motor commands are considered and 6 possible combinations of observation variables (VEL[t], POS[t] and

in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1440-1445, Edinburgh, 2005.

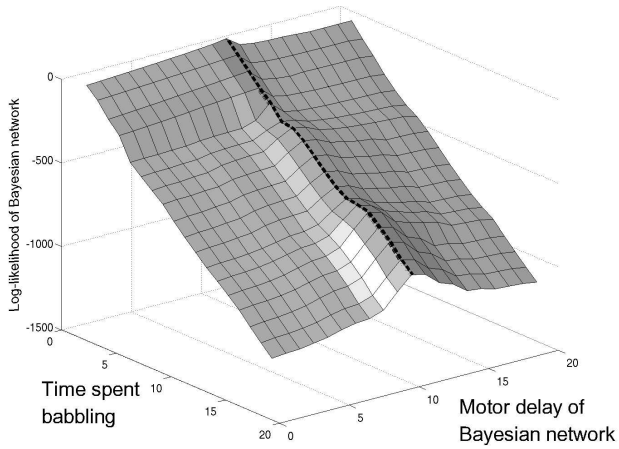

Figure 9: *How the log-likelihoods of the Bayesian networks using VEL[t] as the observation node evolve over the course of a typical experiment. The maximum log-likelihood is for the Bayesian network with a motor delay of 11 time steps, or 550 ms*

SIZE[t], for each gripper), the search space contains 120 possible models. Figure 9 shows how the log-likelihoods for the Bayesian networks with varying motor delays and the velocity observation change over time. As the figure shows, the motor delay which maximise the log-likelihood, *d*, was 11 time-steps, which is about 550 ms. Following the search, the learnt forward model for the grippers is shown in figure 10. The motor command was learnt to be M[*t*-11], and the best observation was learnt to be the velocity of the grippers. This was correct since the motor commands do control the grippers' velocities.

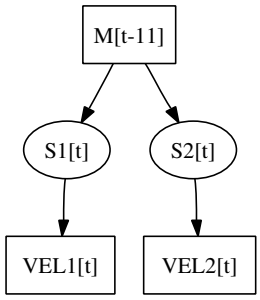

Figure 10: *The learnt forward model of the grippers.*

5 Using the Bayesian network as a forward model

Once the Bayesian network in figure 10 has been learnt, it can be used as a forward model to give a prediction of the consequences of the motor command. For the forward models of the grippers, the predicted output is the velocity of the grippers, calculated from the conditional probability distribution: P(VEL[*t*]|M[*t*-4]=*m*). Because the observations here are modelled as Gaussians, the most likely values will be the

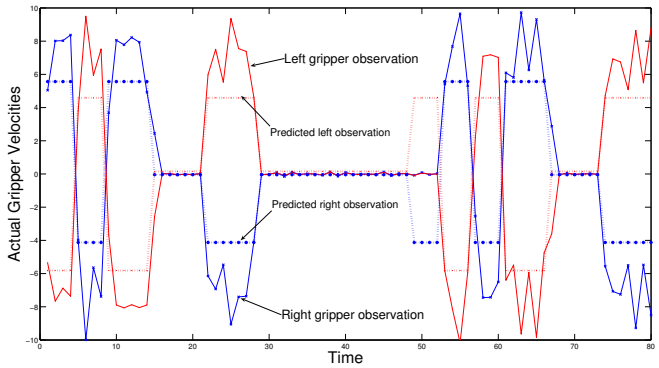

Figure 11: *Evaluating the predictive abilities of a learnt forward model. The predicted gripper movements (dashed) are close to the actual ones (solid). The differences are principally because the variation in gripper velocities for each of the three motor commands are modelled as noise in the forward model.*

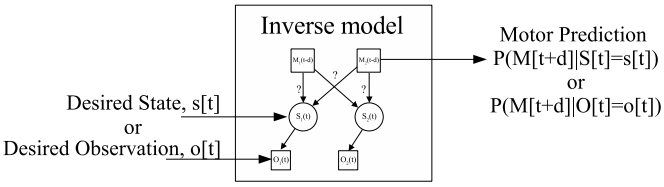

Figure 12: *Using the learnt Bayesian network as an inverse model. Evidence is supplied to the observations or the state, and the task is to infer the probability distribution of motor commands.*

means of the observations, VEL1[*t*] and VEL2[*t*]. The inference algorithm used to calculate this marginal distribution was the junction-tree algorithm [Pearl, 1988]. To evaluate the quality of the learnt forward model the grippers were babbled in a similar way as before, using a different Markov model. Figure 11 shows how the most likely predicted observation compares with the actual observation. Only the *x*-coordinate predictions are plotted because the grippers' movements are predominantly in this plane. The prediction is shown to be very accurate.

The learnt Bayesian network can also be used as an inverse model by calculating the most likely motor command given the current velocity observation for each time-step, as shown in figure 12. One use of this inverse model is for imitation: by replacing the robot's observations of its own movement with its observations of a human's hand movements in the inverse model, the robot is able to reproduce the motor commands that would be most likely to recreate that human movement in its own gripper system. The same principle is used by Demiris to switch from imitation to recognition, albeit in a multiple paired inverse and forward model system [Demiris, 2002]. Figure 13 shows the results of the imitation experiment: the robot is able to imitate a simple hand waving motion. The human's hand movements are tracked and

in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1440-1445, Edinburgh, 2005.

recognised using the vision system, which is able to correctly track the two hands. The observations are used as evidence for the Bayesian network already learnt by the robot. This network is then used as an inverse model to predict the most likely motor commands that would produce this observation. These motor commands are sent to the robot's motor system enabling it to perform the action that best replicates the observed movement.

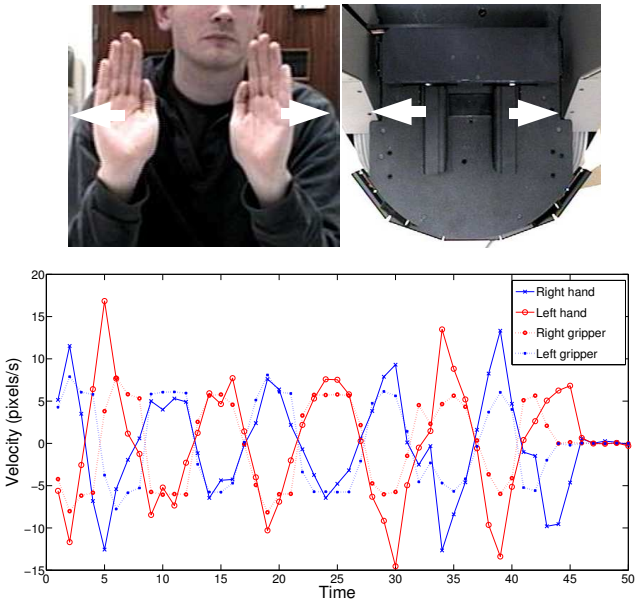

Figure 13: *Imitation using a single inverse model. The top images are corresponding frames from the demonstration (left) and imitating (right) sequences. The graphs show the trajectory of the demonstrating hands, and the corresponding imitating trajectory of the grippers.*

6 Conclusion

Forward models relate a robot's motor commands to their effect on its motor system and the robot's environment. The system presented here allows a robot to autonomously learn forward models of its motor system, using feedback from its vision system, by babbling with its motor commands. By representing the forward model with a Bayesian network, the uncertainty in its prediction can also be represented. An experiment was implemented showing how a robot can learn a forward model of its gripper system. The forward model was subsequently used to allow the robot to imitate a simple 'waving hand' gesture of a human. This imitation occurred without any prior information having to be supplied by a human: the robot learns how to control its motor system, then uses this information to imitate a human gesture. Future work will involve extending the system to learn forward models with more degrees of freedom and interaction with objects.

7 Acknowledgements

This work has been supported through a Doctoral Training Award from the UK's Engineering and Physical Sciences Research Council (EPSRC), and through a bursary from the BBC Research & Development Department. The authors wish to thank Gavin Simmons, Bassam Khadhouri, Matthew Johnson and Paschalis Veskos of the Biologically inspired Autonomous Robots Team (BioART) at Imperial College.

References

[Demiris, 2002] Y. Demiris. 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, 2002.

[Friedman, 1997] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proc. 14th International Conference on Machine Learning*, pages 125–133, 1997.

[Gopnik and Meltzoff, 1998] A. Gopnik and A. N. Meltzoff. *Words, Thoughts, Theories*. MIT Press, 1 edition, 1998.

[Hoppner *et al.*, 1999] Frank Hoppner, Frank Klawonn, Rudolf Kruse, and Thomas Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. John Wiley and Sons Ltd, 1999.

[Jordan and Rumelhart, 1992] M. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, (16):307–354, 1992.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[Meltzoff and Moore, 1997] A. N. Meltzoff and M. K. Moore. Explaining facial imitation: A thoretical model. *Early Development and Parenting*, (6):179–192, 6 1997.

[Neapolitan, 2004] Richard E. Neapolitan. *Bayesian Structure Learning*. Prentice Hall, 2004.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Roy and Thrun, 1999] Nicholas Roy and Sebastian Thrun. Online self-calibration for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[Tong and Koller, 2000] S. Tong and D. Koller. Active learning for parameter estimation in bayesian networks. *Advances in Neural Information Processing Systems (NIPS)*, 12 2000.

[Tong and Koller, 2001] S. Tong and D. Koller. Active learning for structure in bayesian networks. *Seventeenth International Joint Conference on Artificial Intelligence*, pages 863–869, 2001.

[Wolpert *et al.*, 1998] D. M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998.