Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

Active learning of probabilistic forward models in visuo-motor development

Anthony Dearden and 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 both robots and humans to predict the sensory consequences of their motor actions. To learn its own forward models a robot needs to experiment with its own motor system, in the same way that human infants need to babble as a part of their motor development. In this paper we investigate how this babbling with the motor system can be influenced by the forward models' own knowledge of their predictive ability. By spending more time babbling in regions of motor space that require more accuracy in the forward model, the learning time can be reduced. The key to guiding this exploration is the use of probabilistic forward models, which are capable of learning and predicting not just the sensory consequence of a motor command, but also an estimate of how accurate this prediction is. An experiment was carried out to test this theory on a robotic pan tilt camera.

1 Introduction

Forward models enable both robots and humans to predict the sensory consequences of their motor actions [Jordan and Rumelhart, 1992, Wolpert and Flanagan, 2001]. This is extremely useful for robotics as it allows the robot to simulate the effects of its actions internally before physically executing them. Being able to simulate multiple possible actions allows the robot to choose the most appropriate command for a particular task, for example imitation [Demiris and Johnson, 2003]. Practically any environment a robot operates 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 ability or desire of its programmer. A truly autonomous robot, therefore, needs to be able to learn and adapt its own forward models.

The idea of learning a model of an unknown system is explored extensively in the field of system identification [Ljung, 1987]. In system identification, the task of choosing experiments and interventions to perform on the unknown system is the role of the human designing the control system. Here, however, we want this process to be automated - the robot should essentially design its own experiments. The idea of a robot as a scientist provides some interesting analogies with the theories of learning in human infants presented by Gopnik [Gopnik et al., 2004], who uses Bayesian networks to model how infants actively form and test causal models of the world. Meltzoff discussed 'body babbling' as a method used by human infants to learn and adapt control of their motor system [Meltzoff and Moore, 1997].

By using as little prior information as possible, we want the robot to learn about its own motor system. This knowledge it gains about its motor system is stored in the form of a forward model. Previous work on learning forward models has looked at how a robot can develop an internal representation of the state of the world with information from it's vision system [Dearden and Demiris, 2005]. The forward model for predicting the state was learnt and represented probabilistically using a Bayesian network. The training data was provided by random babbling of motor commands to produce the corresponding set of sensor data to train the model. The work here expands on this by allowing the exploration, or babbling, of the motor system to be driven by the estimated prediction accuracy of multiple competing forward models. By spending more time babbling with motor commands which the forward models are worse at predicting, the forward models can be more rapidly learnt and used.

Active exploration of the environment by a robot to learn or adapt models has been attempted previously, in [Lipson and Bongard, 2004]. Using multiple inter-

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

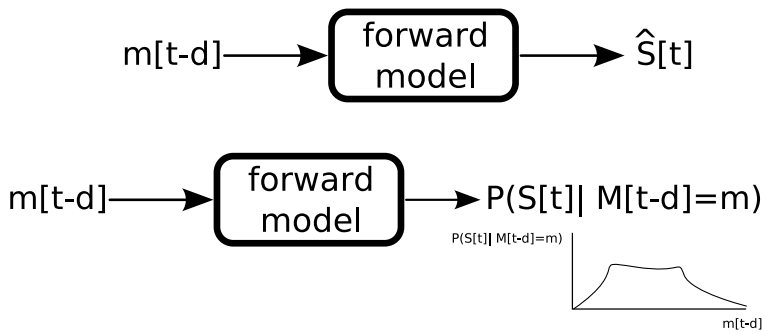

Figure 1: *A forward model and a probabilistic forward model for motor command* m[t], *sensor prediction,* $\hat{S}$*, and motor delay d.*

nal models generated and adapted using a genetic algorithm, their exploration-estimation algorithm uses a two phase process of choosing motor commands to best discriminate between potential models. The exploration is not driven by the prediction error as in this paper, but by choosing interventions which will maximally differentiate between the different internal models. The idea of 'adaptive curiosity' is used in [Oudeyer et al., 2005] to guide a robot to learn how to interact with its environment. The robot is made to focus on situations that are progressively harder for it to predict.

2 An architecture for learning and representing forward models

The system proposed here for learning the model of a robot's motor system is based on using multiple probabilistic forward model 'primitives'. Active learning is used to decide how motor commands should be chosen by each individual forward model primitive, and selected from the multiple possible commands requested by the forward models.

2.1 Why *probabilistic* forward models?

All forward models are wrong, but some are useful[1]. A forward model will not be able to completely accurately model a robot's motor system - errors will occur in predictions because of insufficient or noisy training data or the necessarily simplified internal representations of the model. The system which is being modelled may itself be stochastic. To overcome this inaccuracy, it makes sense for a forward model to include information regarding not just its prediction, but how accurate it expects that prediction to be. This inaccuracy can be modelled by having the forward model learn not just a prediction for a given motor command and state, but a joint probability distribution across the inputs to the forward model and its predicted outputs. The output of a probabilistic forward model is thus a conditional probability distribution for a particular motor command, *m*, and state, *s*, at time *t* : $P\left(S\left[t\right] \mid M\left[t-d\right]=m\right)$, as shown in Figure 1. The other parameter, *d*, is used to model the delay in the motor system - in any real system, there will be a delay between a motor command being executed and its effects being measured at its sensors. For a forward model to be useful in this situation it must model (and learn) this delay.

The advantage of a probabilistic representation of prediction is that, instead of predicting a specific outcome, the prediction will be of a range of possible outcomes, each weighted with a particular likelihood. The forward model essentially has knowledge about its own ability to predict. Any control system using the forward model will receive not just one prediction, but a probability distribution. This provides the control system with more information about the predicted consequences of its actions. This extra information is also useful for guiding the motor control during the learning process. The disadvantage of using a probabilistic representation is that more training data may be required. This is not as much of a disadvantage as it would be in a typical machine learning situation because the data set is not limited - the robot has active control over the system it is trying to model, so can easily acquire training data.

To overcome the trade-off between the complexity of the modelled conditional probability distribution and the amount of training data - and therefore time - required to train it, the normal distribution was

[1] A modification of a quote attributed to George EP Box

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

used. The forward model therefore needs to learn and represent two functions: $\hat{S}(m)$ - the estimated mean of the sensor value as a function of the motor input(s) $\hat{\sigma}_S(m)$, the estimated standard deviation. The output distribution as a function of the motor command, *m*, is therefore $P\left(S\left[t\right] \mid M\left[t-d\right]=m\right) \sim N\left(\hat{S}\left(m\right), \hat{\sigma}\left(m\right)\right)$. Both these functions can be estimated with any appropriate function approximator that can be learnt online. In the experiments here, both radial basis functions and conditional probability tables were used.

2.2 Why *multiple* forward models?

The idea of using multiple forward models has been used in both robotics for imitation [Demiris, 2002], and in neuroscience to model motor skill learning in humans [Wolpert et al., 2003]. In these architectures, the multiple forward models are used together with inverse models to achieve higher level control. In this work, however, we are just interested in learning the forward model that can be used by these systems.

Using multiple primitive forward models to model a system is similar to the mixture of experts idea introduced by [Jacobs et al., 1991]. As the forward models are probabilistic and represent causal connection between the random variables for motor command, *M*, and predicted output, *S*, the forward models make up a Bayesian network [Pearl, 1988]. The forward models are the conditional probability distributions connecting random variables. Splitting the forward model into a distributed system using multiple, simpler forward models has numerous advantages over using a single forward model

- The learnt structure represents causal structure of the robot's motor system. This means the learning process requires less data (and is therefore faster) because unnecessary connections between motors and sensors are not learnt. The robot also has an internal representation of the higher level causal structure of its motors system.
- Robots have different kinds of motor commands and sensors (e.g. discrete or continuous). The appropriate internal representation for the forward model may be different depending on the nature of these. Using multiple forward models allows several different types of function approximators to be used simultaneously.

Figure 2 shows a comparison between a single and multiple primitive forward models.

2.3 Active learning and babbling

In a typical machine learning situation, it is assumed that a set of data representing samples from an underlying function or probability distribution is available. The task is to learn a function or distribution which approximates this distribution. The situation with a robot is different in two ways. Firstly, the process is performed online as opposed to in a batch - data is continuously received and the learnt forward models should be continuously adapted. Secondly, and most importantly, the robot has active control over the inputs it can send to its as yet unknown motor system. The situation where the learner has the ability to select some of the data is referred to as active learning [Hasenjager and Ritter, 2002, Tong and Koller, 2000]. The principal benefit of this is that the data can be selected either to speed up the learning process, or to optimise the learnt model to be most useful for a particular task. For example a robot could concentrate on learning particular forward models that would be needed to imitate a specific task.

The use of multiple competing forward models fits well into the concept of active learning, as each primitive forward model can now compete not just to offer the best prediction, but also to get control over the motor system to provide itself with training data. This does, however, complicate the situation somewhat. As well as the problem of how each forward model chooses a motor command or set of motor commands to be sent to the motor system, there is the important issue of how to choose which of the forward models should be given control of the motor system at any particular time.

This problem has many similarities to attention mechanisms studied in robotics [Khadhouri and Demiris, 2005, Demiris and Khadhouri, to appear], which investigate the allocation of processing resources. In contrast, the task here is to control the allocation of *motor* resources. In this paper the approach taken to guide the babbling is to allow each forward model to suggest a particular motor degree of freedom and value to babble with. The probability of a particular motor command being chosen by a forward model is proportional to the estimated standard deviation of the forward model in that region of motor space, $\hat{\sigma}_S(m)$. Therefore, the forward model is more likely to pick a motor command that it estimates has high prediciton error. Several motor commands will be requested simultaneously, one for each forward model. The learning system currently chooses a forward model at random to ensure that each forward model is given the opportunity to control the motor system.

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

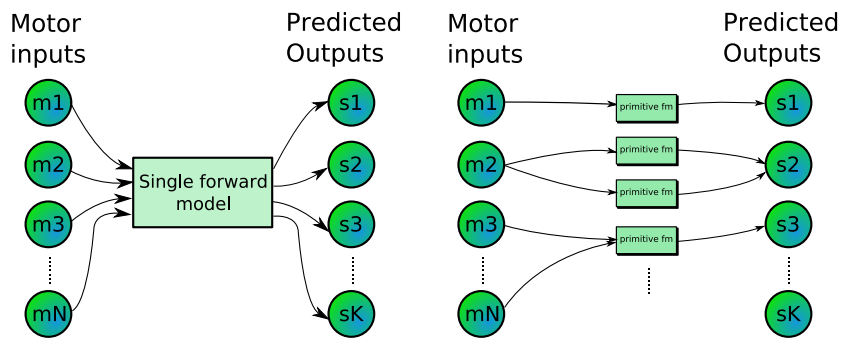

Figure 2: *Using a single forward model (a) or multiple primitive forward models (b)*

3 Stages in learning the forward model

The learning of the forward model needs to be divided into distinct phases. This simplifies learning a complex forward model by learning different aspects of the model's structure or parameters sequentially. The developmental stages represent increasing complexity in the learning and adaptation of forward models, from establishing a causal connection, calculating the time delay, and finally adapting more and more precisely to the causal relationship. The developmental stages used to learn forward models are as follows:

1. Observe and learn a steady state model of the sensors

   In this first stage of learning, the robot does not actually interact with the environment - it simply learns the statistics of the sensor data $P(S)$ as a normal distribution. This an important preliminary stage to learning any forward model because the robot cannot model how its different motor commands are influencing particular sensors until it has modelled how its sensors behave without any intervention.

2. Try impulse commands to learn time delay, and basic causal structure of the network

   In previous work, the time delay in the motor system was learnt by simultaneously learning multiple forward models with different time delays. The correct time delay was found from the forward model which could best predict the data [Dearden and Demiris, 2005]. Here the time delay is estimated directly by using the learnt models of the sensors. Impulse motor commands are issued to the motor system at time *T*, one degree of freedom at a time. The likelihood of the incoming sensor data, $s[T+t]$ given the sensor model learnt in step one is calculated - i.e. $P(S = s[T+t])$. If this likelihood falls to a low value then it is likely that this motor degree of freedom is influencing this sensor, and that the delay for the influence to occur is *t* discrete time-steps; the threshold likelihood used in the experiments here was 0.001. Thus not only can the motor delay be learnt, but some initial information about the causal structure of the forward model is learnt - if a motor command does not reduce the likelihood of a sensor model, it is unlikely it can influence it, and therefore this relationship does not have to be modelled.

3. Completely random babbling to learn the range of values for the sensor data

   Function approximators generally need the data to be scaled within a set range, e.g. [0,1]. When sensor data is being received online, and no prior information about it is available, this cannot be done. Therefore, a stage of experimenting with extremes of motor commands to find the extremes in the range of sensor data is necessary. Once this stage of adaptation is complete, the sensor data can be scaled between the calculated minimum and maximum values.

4. Learn steady state model between motor commands and sensors, using guided babbling

   The guided babbling in this stage happens as described in section 2.3. Because we are currently only interested in learning steady state models, learning is paused from the issuing of a motor until the sensor system has reached a steady state.

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

4 Experiment & results

The experiments here were carried out using the pan-tilt unit on an Activmedia Peoplebot[2]. The sensor data used were the properties of the most salient coloured object in the scene - its position, width, height and angle of rotation. The object is located from the thresholded camera image in hue space, and tracked using the Camshift algorithm [Bradski, 1998]. The first and second stage of the learning process identified the delay for both the pan and tilt motor commands to be 5 time-steps, or 333ms. As shown in Figure 3 , it also learnt that, whilst the co-

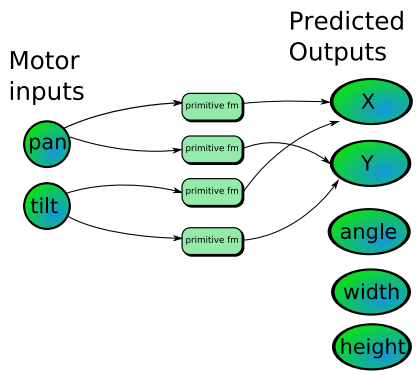

Figure 3: *The primitive forward model structures learnt from sending impulse commands to the motor system.*

ordinates of the object in the scene were affected by the pan and tilt commands, the size and angle of the object were not affected. By learning this causal relationship early on, the robot has thus reduced the number of models it has to initially learn from ten to four.

The evolution of the prediction errors, from a typical experiment, for the four forward models using guided babbling are shown in Figure 4a . The prediction error is as the sum of the variance estimation over all motor commands, $\int \hat{\sigma_s}(m)\, dm$. This can be compared with the evolution of the prediction errors when random motor commands are chosen, as shown in Figure 4b. Converging to accurate models takes significantly longer in this case.

If the camera had been mounted perfectly straight then the *pan* motor command would have no effect on the y-coordinate of the object, and similarly for the *tilt* command and the x-coordinate. However, since the camera is at a slight angle, there is a slight dependence between these. It is interesting to note that whilst the forward models linking the *pan* command to the *y*-coordinate and the *tilt* command to the *x*-coordinate do converge to a particular model, they are much less accurate at predicting than the *pan*-to-*x*-coordinate and *tilt*-to-*y*-coordinate forward models, as one would expect.

Part of the evolution of the *pan*-to-*x*-coordinate forward model's mean prediction, $\hat{S}(m)$, is shown in Figure 5. As expected, the model learnt is a linear one - the position of the object, *X*, is proportional the the *pan* command. Of particular interest is the prediction of a low valued motor command, as shown by the bold line. Because the estimated error in prediction is initially high for this motor command, more time is spent babbling in this region, and hence it converges to a more accurate model.

5 Conclusions and future work

In this paper, we investigated how the learning of forward models for a robot could be made faster by allowing the forward models to guide the exploration of the motor space with guided babbling. The results show that accurate models can be learnt more quickly if the errors in the predictions of the forward models are used to guide which region of motor space is explored. Future work will involve investigating this idea further, by looking at how the motor requests from each individual model should be allocated. Important factors in this decision include:

- How to cope with many more degrees of freedom
- How well a forward model is predicting
- How much data the forward model has previously been allowed
- What is the goal of the babbling - to learn a model as fast as possible or as accurate as possible for a particular task?
- How many primitive forward models want access to the same region of motor space

We are also investigating how the primitive forward models can be improved to represent and adapt to dynamic environments by adding another stage to the learning process.

References

G. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *4th*

[2]http://www.activmedia.com

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

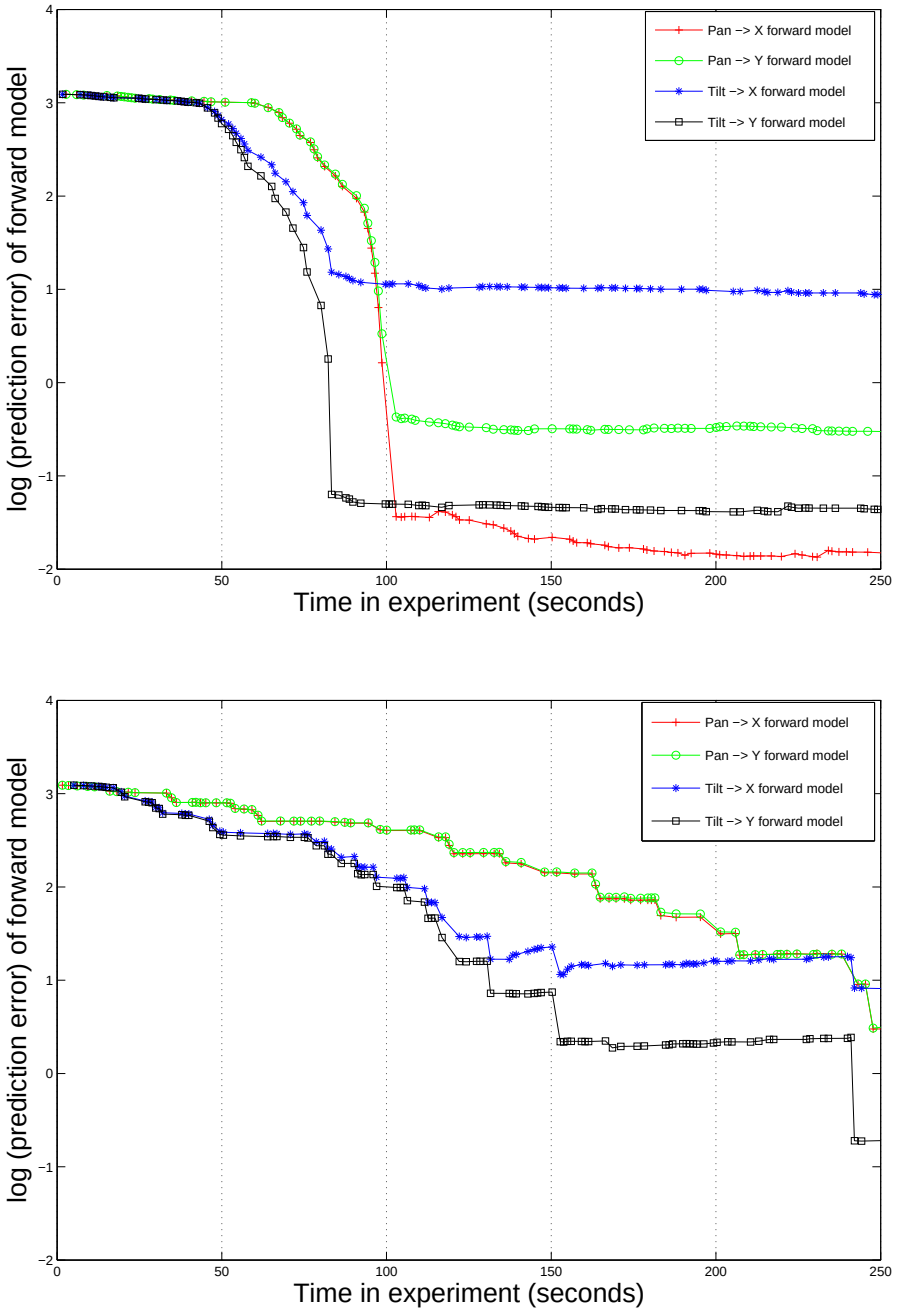

Figure 4: *The evolution of the prediction errors of the four of the forward models created when using guided babbling (a) and random babbling (b).*

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

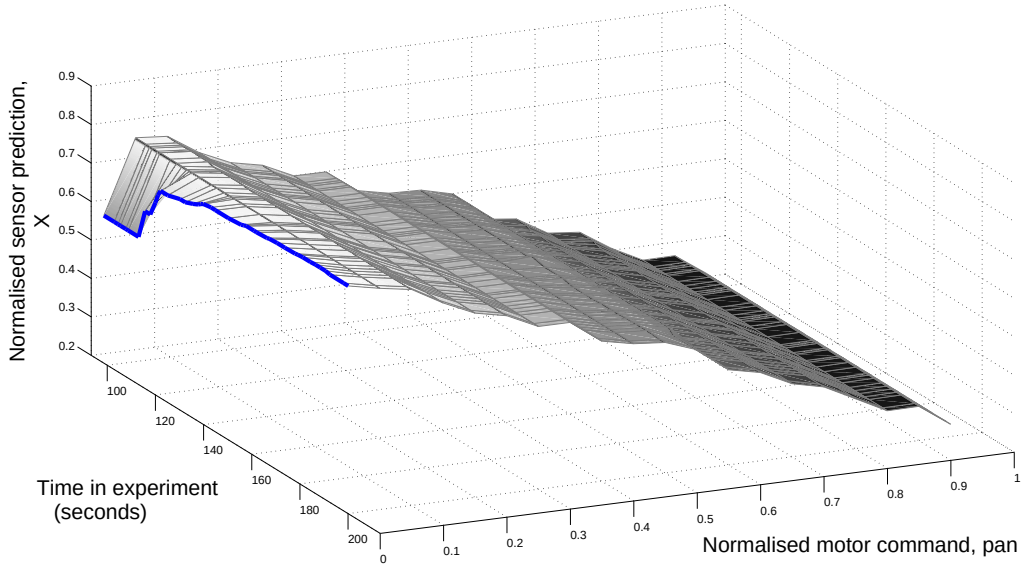

Figure 5: *Evolution of the pan -> x forward model during the experiment. The bold line shows the evolution of the motor command prediction where there is high error. Because more time is spent babbling with this motor command, the prediction converges quickly to a more correct one.*

IEEE Workshop on Applications of Computer Vision (WACV'98), page 214. IEEE, 1998.

A. Dearden and Y. Demiris. Learning Forward Models for Robotics. In *Proceedings of IJCAI 2005*, pages 1440–1445, 2005.

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.

Y. Demiris and M. Johnson. Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. *Connection Science*, 15(4), 12 2003.

Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition (hammer). *Robotics and Autonomous Systems Journal*, to appear.

A. Gopnik, C. Glymour, D. Sobel, and D. Danks. A theory of causal learning in children: Causal maps and Bayes nets. *Psychological Review*, pages 1–30, 1 2004.

M. Hasenjager and H. Ritter. Active learning in neural networks. *New learning paradigms in soft computing*, pages 137–169, 2002.

R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3:79–87, 1991.

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

B. Khadhouri and Y. Demiris. Compound effects of top-down and bottom-up influences on visual attention during action recognition. In *Proceedings of IJCAI-2005*, pages 1458–1463, 2005.

H. Lipson and J. Bongard. An exploration-estimation algorithm for synthesis and analysis of engineering systems using minimal physical testing. In *Proceedings of DETC04*, 2004.

L. Ljung. *System Idenetification, Theory for the user*. Prentice Hall, 1987.

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

P. Oudeyer, F. Kaplan, V. Hafner, and A. Whyte. The playground experiment: Task-independent development of a curious robot. In *proceedings of the*

Preprint version; final version available at http://www.aisb.org.uk/
AISB'06: Adaptation in Artificial and Biological Systems (2006), vol: 1, pp: 176-183

AAAI Spring Symposium Workshop on Developmental Robotics, 2005.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

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

D. M. Wolpert and J. R. Flanagan. Motor Prediction. *Current Biology*, 11(18), 2001.

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