Chapter 1

ROBOT COMPETENCE DEVELOPMENT BY CONSTRUCTIVE LEARNING

Q. MENG[1], M.H. LEE[2] AND C.J. HINDE[1]
[1]DEPARTMENT OF COMPUTER SCIENCE, LOUGHBOROUGH UNIVERSITY,
LE11 3TU, UK. Q.MENG@LBORO.AC.UK
[2]DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF WALES,
ABERYSTWYTH, SY23 3DB, UK.

Abstract This paper presents a constructive learning approach for developing sensor-motor mapping in autonomous systems. The system's adaptation to environment changes is discussed and three methods are proposed to deal with long term and short term changes. The proposed constructive learning allows autonomous systems to develop network topology and adjust network parameters. The approach is supported by findings from psychology and neuroscience especially during infants cognitive development at early stages. A growing radial basis function network is introduced as a computational substrate for sensory-motor mapping learning. Experiments are conducted on a robot eye/hand coordination testbed and results show the incremental development of sensory-motor mapping and its adaptation to changes such as in tool-use.

Keywords: Developmental robotics, biologically inspired systems, constructive learning, adaptation

1. Introduction

In many situations such as home services for elderly and disabled people, artificial autonomous systems (e.g. robots) need to work for various tasks in an unstructured environment, system designers cannot anticipate every situation and program the system to cope with them. This is different from the traditional industrial robots which mostly work in structured environments and are programmed each time for a specific task. Autonomy, self-learning and organizing, and adapting to environment changes are crucial for these artificial systems to successfully fulfil various challenging tasks. Traditional controllers for intelligent systems are designed by hand, and they do not have such flexibility and adaptivity. General cognitivist approach for cognition is based on symbolic information processing and representation, and does not need to be embodied and physically interact with the environment. Most cognitivist-based artificial cognitive systems rely on the experience from human designers.

Human beings and animals face similar problems during their development of sensor-motor coordination, however we can tackle these problems without too much effort. During human cognitive development, especially at the early stages, each individual undergoes changes both physically and mentally through interaction with environments. These cognitive developments are usually staged, exhibited as behavioural changes and supported by neuron growth and shrinking in the brain. Two kinds of developments in the brain support the sensory-motor coordination: quantitative adjustments and qualitative growth [Shultz, 2006]. Quantitative adjustments refer to the adjustments of the synapse connection weights in the network and qualitative growth refers to the changes of the topology of the network. Inspired by developmental psychology especially Piaget's sensory-motor development theory of infants [Piaget, 1952], developmental robotics focuses on mechanisms, algorithms and architectures for robots to incrementally and automatically build their skills through interaction with their environment [Weng et al., 2001]. The key features of developmental robotics share similar mechanisms with human cognitive development which include learning through sensory-motor interaction; scaffolding by constraints; staged, incremental and self-organizing learning; intrinsic motivation driven exploration and active learning; neural plasticity, task transfer and adaptation. In this paper, we examine robot sensory-motor coordination development process at early stages through a constructive learning algorithm. Constructive learning which is inspired by psychological constructivism, allows both quantitative adjustments and qualitative network growth to support the developmental learning process. Most static neural networks need to predefine the network structure and learning can only affect the connection weights, and they are not consistent with developmental psychology. Constructive learning is supported by recent neuroscience findings of synaptogenesis and neurogenesis occurring under pressures to learn [Quartz and Sejnowski, 1997, Shultz et al., 2007]. In this paper, a self-growing radial basis function network (RBF) is introduced as the computational substrate, and a constructive learning algorithm is utilized to build the sensory-motor coordination development. We investigate the plasticity of the network in terms of self-growing in network topology (growing and shrinking) and adjustments of the parameters of each neuron: neuron position, the size of receptive field of each neuron, and connection weights. The networks adaptation to systems changes is further investigated and demonstrated by eye/hand coordination test scenario in tool-use.

2. Sensory-motor mapping development via constructive learning

In order to support the development of sensor-motor coordination, a self-growing RBF network is introduced due to its biological plausibility. There exists very strong evidence that humans use basis functions to perform sensorimotor transformations [Pouget and Snyder, 2000], Poggio proposed that the brain uses modules as basis components for several of its information processing subsystems and these modules can be realized by generalized RBF networks [Poggio and Girosi, 1990, Poggio, 1990].

There are three layers in the RBF network: input layer, hidden layer and output layer. The hidden layer consists of radial basis function units (neurons), the size of receptive field of each neuron varies and the overlaps between fields are different. Each neuron has its own centre and coverage. The output is the linear combination of the hidden neurons.

A RBF network is expressed as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{a}_0 + \sum_{k=1}^{N} \mathbf{a}_k \phi_k(\mathbf{x}) \tag{1.1}$$

$$\phi_k(\mathbf{x}) = \exp(-\frac{1}{\sigma_k^2} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2) \tag{1.2}$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_{N_o}(\mathbf{x}))^T$ is the vector of system outputs, N_o is the number of outputs and $\mathbf{X}$ is the system input. $\mathbf{a}_k$ is the weight vector from the hidden unit $\phi_k(\mathbf{x})$ to the output, N is the number of radial basis function units, and $\boldsymbol{\mu}_k$ and σ_k are the kth hidden unit's center and width, respectively.

Why constructive learning?

According to Shultz [Shultz, 2006, Shultz et al., 2007], in addition to that constructive learning is supported by biological and psychological findings, there are several advantages of constructive learning over static learning: first, constructive-network algorithms learn fast (in polynomial time) compared with static learning (exponential time), and static learning maybe never solve some problems as the designer of a static network must first find a suitable network topology. Second, constructive learning may find optimal solutions to the bias/variance tradeoff by reducing bias via incrementally adding hidden units to expand the network and the hypothesis space, and by reducing variance via adjusting connection weights to approach the correct hypothesis. Third, static learning cannot learn a particular hypothesis if it has not been correctly represented, a network may be too weak to learn or too powerful to generalize. Constructive learning avoids this problem because its network growth enables it to represent a hypothesis that could not be represented previously with limited network power.

Topological development of the sensory-motor mapping network

During the development of sensory-motor mapping network, two mechanisms exist: topological changes of the mapping network and network parameter adjustments. The qualitative growth of the sensory-motor mapping network depends on the novelty of the sensory-motor information which the system obtained during its interaction with the environment in development, the growth is incremental and self-organizing. The sensory-motor mapping network starts with no hidden units, and with each development step, i.e. after the system observes the consequence of an action, the network grows or shrinks when necessary or adjusts the network parameters accordingly. The network growth criteria are based on the novelty of the observations, which are: whether the current network prediction error for the current learning observation is bigger than a threshold, and whether the node to be added is far enough from the existing nodes in the network: $\|\mathbf{e}(t)\| = \|\mathbf{y}(t) - \mathbf{f}(\mathbf{x}(t))\| > e_1$, $\|\mathbf{x}(t) - \boldsymbol{\mu}_r(t)\| > e_3$. In order to ensure smooth growth of the network the prediction error is checked within a sliding window: $\sqrt{\sum_{j=t-(m-1)}^{t} \frac{\|\mathbf{e}(j)\|^2}{m}} > e_2$, where, $(\mathbf{x}(t), \mathbf{y}(t))$ is the learning data at tth step, and $\boldsymbol{\mu}_r(t)$ is the centre vector of the nearest node to the current input $\mathbf{x}(t)$. m is the length of the observation window. If the above three conditions are met, then a new node is inserted into the network with the following parameters: $\mathbf{a}_{N+1} = \mathbf{e}(t),\ \boldsymbol{\mu}_{N+1} = \mathbf{x}(t),\ \sigma_{N+1} = k\,\|\mathbf{x}(t) - \boldsymbol{\mu}_r(t)\|$, where, k is the overlap factor between hidden units.

The above network growth strategy does not include any network pruning, which means the network size will become large, some of the hidden nodes may not contribute much to the outputs and the network may become overfit. In order to overcome this problem, we use a pruning strategy as in [Lu et al., 1998], over a period of learning steps, to remove those hidden units with insignificant contribution to the network outputs.

Let o_{nj} be the jth output component of the nth hidden neuron, $o_{nj} = a_{nj}\exp(-\frac{\|\mathbf{x}(t)-\boldsymbol{\mu}_n\|^2}{\sigma_n^2}),\ r_{nj} = \frac{o_{nj}}{max(o_{1j}, o_{2j}, \cdots, o_{Nj})}$

If $r_{nj} < \delta$ for M consecutive learning steps, then the nth node is removed. δ is a threshold.

Parameter adjustments of the sensory-motor mapping network

There are two types of parameters in the network, the first type of parameter is the connection weights; the second is parameters of each neuron in the network: the position and the size of receptive field of each neuron. A simplified node-decoupled EKF (ND-EKF) algorithm was proposed to update the parameters of each node independently in order to speed up the process. The parameters of the network are grouped into $N_o + N$ components. The first N_o groups are the weights, $\mathbf{w}_k = [a_{0k}, a_{1k}, \cdots, a_{Nk}]^T, k = 1, 2, \cdots, N_o$ (a_{ij} is the weight from ith hidden node to jth output); and the rest N groups are the parameters of hidden units' parameters: $\mathbf{w}_k = [\boldsymbol{\mu}_k^T, \sigma_k]^T, k = 1, 2, \cdots, N$. The superscript T stands for transpose of a matrix.

So for kth parameter group at tth learning step, ND-EKF is given by:

$$\mathbf{w}_k(t) = \mathbf{w}_k(t-1) + \mathbf{K}_k(t)\mathbf{e}_k(t) \tag{1.3}$$

where

$$\mathbf{e}_k(t) = \begin{cases} y_k(t) - f_k(\mathbf{x}(t)) & k = 0, 1, 2, \cdots, N_o \\ \mathbf{y}(t) - \mathbf{f}(\mathbf{x}(t)) & k = N_o + 1, \cdots, N_o + N \end{cases} \tag{1.4}$$

and $\mathbf{K}_k(t)$ is the kalman gain, $y_k(t)$ is the kth component of $\mathbf{y}(t)$ in training data $(\mathbf{x}(t), \mathbf{y}(t))$, $\mathbf{B}_k(t)$ is the submatrix of derivatives of network outputs with respect to the kth group's parameters at tth learning step. $\mathbf{R}_k(t)$ is the variance of the measurement noise, and is set to be $diag(\lambda)$ (λ is a constant) in this paper. q is a scalar that determines the allowed random step in the direction of the gradient vector.

In our algorithm, an extended Kalman filter is used to adjust the systems's parameters. There may exist a similar mechanism in our brain. Recent research findings has found evidences that Kalman filtering occurs in visual information processing [Rao and Ballard, 1997, Rao and Ballard, 1999], motor coordination control [Todorov and Jordan, 2002], and spatial learning and localization in the hippocampus [Bousquet et al., 1998, Szirtes et al., 2005]. In hippocampus studies, a Kalman filtering framework has been mapped to the entorhinal-hippocampal loop in a biologically plausible way [Bousquet et al., 1998, Szirtes et al., 2005]. According to the mapping, region CA1 in the hippocampus holds the system reconstruction error signal, and the internal representation is maintained by Entorhinal Cortex (EC) V-VI. The output of CA1 corrects the internal representation, which in turn corrects the reconstruction of the input at EC layers II-III. O'Keefe also provided a biologically plausible mechanism by which matrix inversions might be performed by the CA1 layer through an iterated update scheme and in conjunction with the subiculum [O'Keefe, 1989]. In addition, the matrix inversion lemma has been widely used in computational neuroscience [Huys et al., 2007].

3. Adaptation of sensory-motor mapping

Two kinds of changes in our daily life may require the learned sensory-motor mapping to update: short term changes and long term changes. For the short term, humans may just reuse learned knowledge and quickly adjust some parameters to adapt to the environment changes. But for the longer term, after an adult is trained in a special environment or for a special tasks for a long time, they may grow new neurons to gain new skills, and to enhance the already acquired knowledge. Examples of these two kinds of changes can be found during human development, the kinematics of limbs and bodily structures are not fixed during human growth but may change, either slowly over long periods during growth and bodily maturation, or rapidly such as when we use tools to extend the reach or function of our manipulation abilities. It has been discovered that infants learn and update sensorimotor mappings by associating spontaneous motor actions and their sensory consequences [Piaget, 1952]. It takes a relatively long time to build up the mapping skills, which involves neuron growth processes in the brain to support the sensorimotor transformation. After an adult has gained the basic skills, they can quickly adapt to different situations, for example, an adult can quickly adapt to the use of a pointer to point to a seen target. This indicates that after rapid structural changes we do not learn new sensorimotor skills from scratch, rather we reuse the existing knowledge and simply (and quickly) adjust some parameters. Maguire at al [Maguire et al., 2000] studied the structural changes in the hippocampi of licensed London tax drivers. They found that taxi drivers had a significantly greater volume in the posterior hippocampus, whereas control subjects showed greater volume in the anterior hippocampus. Maguire's study suggests that the human brain grows or shrinks to reflect the cognitive demands of the environment, even for adults.

In autonomous systems, some parameters may gradually change after a long time use, the systems need to adapt to these changes automatically. Autonomous systems have additional situations where structures may change suddenly, these may be unintentional, for example when damage occurs through collisions, or by design when a new tool is fitted to the arm end-effector. For these reasons it is important for autonomous systems in unstructured environments to have the ability to quickly adjust the existing mapping network parameters so as to automatically re-gain the eye/hand coordination skills. We note that humans can handle this problem very well. Recent neurophysiological, psychological and neuropsychological research provides strong evidence that temporal, parietal and frontal areas within the left cerebral hemisphere in humans and animals are involved and change during activities where the hand has been extended physically, such as when using tools [Imamizu et al., 2000, Johnson-Frey, 2004, Maravita and Iriki, 2004, Hihara et al., 2006, Hihara et al., 2003]. Japanese macaque monkeys were trained to use a rake to pull food closer, which was originally placed beyond the reach of their hands [Hihara et al., 2006, Hihara et al., 2003]. The researchers found that, in monkeys trained in tool-use, a group of bimodal neurons in the anterior bank of the intraparietal sulcus, which respond both to somatosensory and visual stimuli related to the hand, dynamically altered their visual receptive field properties (the region where a neuron responds to certain visual stimuli) during training of the tool-use.

In this paper, we develop approaches of adapting to environments, and more specifically, different robot limb sizes in our experiments, were investigated and compared. All these adaptation skills are usually not available in commercial calibration-based eye/hand mapping systems.

In our plastic RBF network for robotic eye/hand mapping, the knowledge learned for the mapping is stored in the network in terms of the number of neurons, their positions and sizes of receptive fields, and the node weights. In order to quickly adapt to structural changes of the robotic system, this knowledge needs to be reused in some way rather than setting up the network again from empty. In this paper, we considered three methods for such adaptation, all of them reuse the learned knowledge by adjusting the learned network:

1 Full adjustment of the learned network after a structural change. This includes network topological changes by adding new hidden nodes or remove existing ones if necessary, and adjusting the following parameters: the centres and widths of the existing nodes, and the weights from the hidden nodes to the outputs.

2 Adjusting the weights of the learned network, removing the insignificant hidden units, but keeping the rest of the hidden units unchanged.

3 Only adjusting the weights, and keeping the hidden unit structure of the learned network completely unchanged.

4. Experimental studies

Experimental system

In this paper, the robot eye/hand coordination is used as a testbed to demonstrate the process of constructive learning and adaptation of the sensory-motor mapping network to the changes. The experimental robot system has two manipulator arms and a motorized pan/tilt head carrying a color CCD camera as shown in Figure 1.1. Each arm can move within 6 degrees of freedom. The whole system is controlled by a PC running XP which is responsible for controlling the two manipulator arms, any tools, the pan/tilt head, and also processing images from the CCD camera and other sensory information. The control program is written in C++.

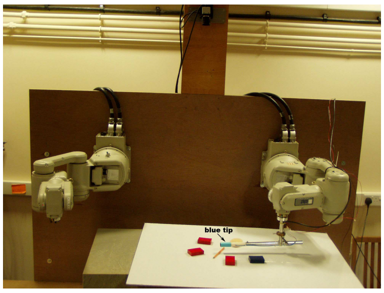
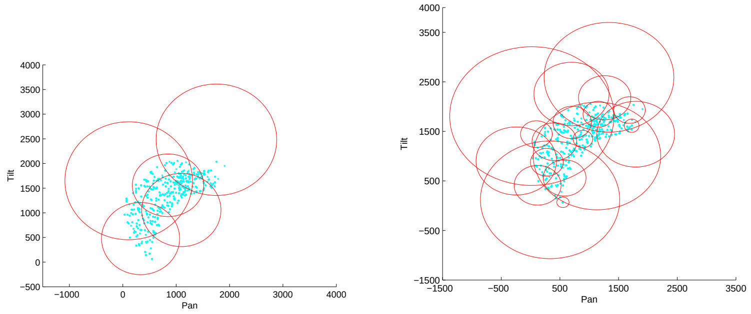


Figure 1.1. Experimental system for developmental coordination learning

In this paper only one of the robot arms was used. In the experiments we commanded the robot arm to move randomly at a fixed height above the table by driving joint 2 and joint 3 of the robot arm. After each movement, if the hand was in the current field of view of the camera, the eye system moved the camera to centre on the end of the robot finger, and then the pan/tilt head position (p, t) and current arm joint values of the two joints used $(j2, j3)$ were obtained to form a training set for the system; otherwise, if the hand tip is out of the view of the camera, this trial was ignored because the eye could not locate the arm end before setting up the mapping between pan/tilt and robot arm. After each trial, the obtained data $(p, t, j2, j3)$ was used to train the mapping network, and this data was used only once. In order to simplify the image processing task of finding the end of the robot finger we marked the finger end with a blue cover. The position of the blue marker could be slid up and down the finger to effectively alter the length of the finger.

Constructive learning and adaptation in tool-use

To illustrate the network topological growth and parameter adjustments in constructive learning, Figure 1.2 gives the structures of the hidden units at the $100th$ learning step and the $1597th$ learning step in eye/hand mapping. The results shows that at the beginning, the system used large neurons to quickly cover the whole space, and later on gradually built the details with smaller neurons when necessary, this let the system achieve more accuracy. This neuron growing process from coarse to fine using different neuron coverages is similar to infant development where the decrease in the size of neural receptive fields in the cortical areas relates to object recognition ability [Westermann and Mareschal, 2004]. Figure 1.2 also demonstrates the changes of position and size of receptive field of each neuron. It should be noted that some neurons are removed in the learning process due to their small contribution to the sensory-motor mapping network.

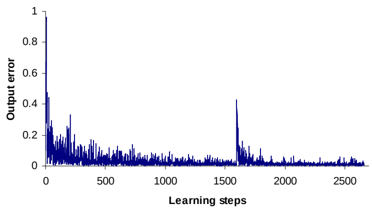


(a) Structure of hidden units at the $100th$ learning step

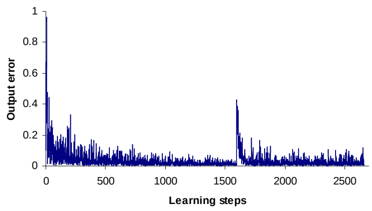
(b) Structure of hidden units at the $1597th$ learning step

Figure 1.2. Distribution of the hidden units and their coverage in eye/hand mapping by RBF with SDEKF. The background points are the input learning points in the pan and tilt space of the camera head, and the circles are the hidden units of the eye/hand mapping network

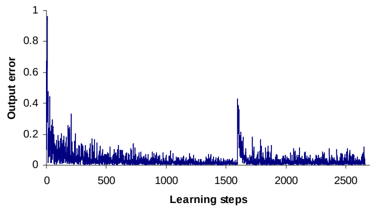
Our next experiment was to test the network's adaptability to sudden changes in the motor-sensory relationship due to structural changes. We chose changes in finger length as a scenario to test this adaptability. Using a variety of tools with different sizes is necessary for a robot system to conduct different tasks, and the eye/hand mapping network's ability to quickly adapt to this change is crucial for the robot to re-gain its eye/hand coordination skills. We have tested three approaches to reusing and adjusting the learned eye/hand mapping network in order to re-gain coordination skills. As a test, at the 1598th trial (a purely arbitrary point) the finger length was changed in size from 27.5cm long to 20.5cm long and we investigated the adaptation of the system to such a sudden change. Figure 1.3(a) shows the output error when all the parameters of the learned network are adjusted, including adding possible nodes, moving node centres, adjusting widths of each node, and updating the weights. Figure 1.3(b) and Figure 1.3(c) show the results of only adjusting the weights and keeping the parameters of the hidden units unchanged, but Figure 1.3(b) used a pruning procedure as described in section 2.0 to remove the insignificant hidden units, while Figure 1.3(c) kept the hidden unit structure completely unchanged. From the results, we can see that all three methods quickly adapt to the sudden change in finger size. The method of adjusting the full network parameters achieved the best result. Although the other two methods did not change the parameters of the hidden units of the learned network, they obtained reasonable small errors. It is important to note that, the third method, which completely reused the original hidden unit structure in the mapping network and only adjusted weights, achieved a quite similar result to the second method with pruning. This may be similar to the approach that adults adopt to handle tool changes. We can quickly adapt to structural changes with little effort, but during such short time-scales we cannot regenerate receptive fields in our brain, and so may only reuse the knowledge already learned and quickly adjust the weights of the existing neurons. But if we are trained to use this tool for a long time, we may improve our operation skills as we might grow new neurons to support the changes as in Figure 1.3(a).



(a) Fully updating the learned network



(b) Updating the weights of the learned network, with pruning



(c) Only updating weights of the learned network, no pruning

Figure 1.3. Adapting to structural change by reusing the learned network in different ways.

Now considering network size, as shown in Figure 1.4, the first method with full updating of all network parameters required by far the largest network, 48 nodes; while the second method removed three hidden units, reducing the network to 16 nodes; the third method kept the original network size, 19 nodes.

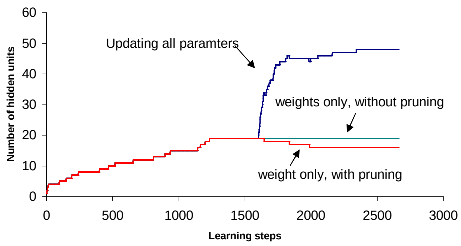


Figure 1.4. The number of hidden units for the three approaches.

We have also studied the staged development in sensory-motor mapping learning process [Lee et al., 2007]. The system constructs sensory-motor schemas in terms of interlinked topological mappings of sensory-motor events, and demonstrates that the constructive learning moves to next stage if stable behaviour patterns emerges.

5. Conclusions

Constructive learning has advantages over static learning in sensory-motor mapping development for autonomous systems. It supports both topological network growth and parameter adjustments, which is supported by findings in psychology and neuroscience. It also has the advantage of adaptation to system changes such as in tool-use. A growing radial basis function network by constructive learning constructs the computational substrate for such sensory-motor mapping development. It forms a platform to examine the relationship between behaviour development and the growth of internal sensory-motor mapping network; the staged and developmental learning process through various constraints in motors and sensors; and active behaviour learning driven by intrinsic motivation. The experimental results on robot eye/hand coordination demonstrate the incremental growth of the mapping network and the system's adaptation to environmental changes.

References

[Bousquet et al., 1998] Bousquet, O, Balakrishnan, K, and Honavar, V (1998). Is the hippocampus a Kalman filter? In *Pacific Symposium on Biocomputing*, pages 655–666, Hawaii.

[Hihara et al., 2006] Hihara, S., Notoya, T., Tanaka, M., Ichinose, S., Ojima, H., Obayashi, S., Fujii, N., and Iriki, A. (2006). Extension of corticocortical afferents into the anterior bank of the intraparietal sulcus by tool-use training in adult monkeys. *Neuropsychologia*, 44(13):2636–2646.

[Hihara et al., 2003] Hihara, S., Obayashi, S., Tanaka, M., and Iriki, A. (2003). Rapid learning of sequential tool use by macaque monkeys. *Physiology and Behavior*, 78:427–434.

[Huys et al., 2007] Huys, Quentin JM, Zemel, Richard S, Natarajan, Rama, and Dayan, Peter (2007). Fast population coding. Neural Computation, 19(2):404-441

[Imamizu et al., 2000] Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Puetz, B., Yoshioka, T., and Kawato, M. (2000). Human cerebellar activity reflecting an acquired internal model of a novel tool. *Nature*, 403:192–195.

[Johnson-Frey, 2004] Johnson-Frey, Scott H. (2004). The neural bases of complex tool use in humans. *Trends in Cognitive Science*, 8(2):71–78.

[Lee et al., 2007] Lee, M. H., Meng, Q., and Chao, F. (2007). Developmental learning for autonomous robots. *Robotics and Autonomous Systems*, 55(9):750–759.

[Lu et al., 1998] Lu, Yingwei, Sundararajan, N., and Saratchandran, P. (1998). Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Transactions on neural networks*, 9(2):308–318.

[Maguire et al., 2000] Maguire, Eleanor A., Gadian, David G., Johnsrude, Ingrid S., Goodd, Catriona D., Ashburner, John, Frackowiak, Richard S. J., and Frith, Christopher D. (2000). Navigation-related structural change in the hippocampi of taxi drivers. *PNAS*, 97(8):4398–4403.

[Maravita and Iriki, 2004] Maravita, A. and Iriki, A. (2004). Tools for the body (schema). *Trends in Cognitive Science*, 8(2):79–86.

[O'Keefe, 1989] O'Keefe, J. (1989). Computations the hippocampus might perform. In Nadel, L., Cooper, L. A., Culicover, P., and Harnish, R. M., editors, *Neural connections, mental computation*. MIT Press, Cambridge, MA.

[Piaget, 1952] Piaget, Jean (1952). *The Origins of Intelligence in Children*. Norton, New York, NY.

[Poggio, 1990] Poggio, Tomaso (1990). A theory of how the brain might work. MIT AI. memo No. 1253.

[Poggio and Girosi, 1990] Poggio, Tomaso and Girosi, Federico (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497.

[Pouget and Snyder, 2000] Pouget, A. and Snyder, L.H. (2000). Computational approaches to sensorimotor transformations. *Nature Neuroscience supplement*, 3:1192–1198.

[Quartz and Sejnowski, 1997] Quartz, S.R. and Sejnowski, T.J. (1997). The neural basis of cognitive development: A constructivist manifesto. *Brain and Behavioral Sciences*, 20:537–596.

[Rao and Ballard, 1997] Rao, R. and Ballard, D. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9(4):721–763.

[Rao and Ballard, 1999] Rao, R. and Ballard, D. (1999). Predictive coding in the visual cortex. *Nature Neuroscience*, 2(1):79–87.

[Shultz, 2006] Shultz, T. R (2006). Constructive learning in the modeling of psychological development. In Munakata, Y. and Johnson, M. H., editors, *Processes of change in brain and cognitive development: Attention and performance XXI*, pages 61–86. Oxford: Oxford University Press.

[Shultz et al., 2007] Shultz, T. R., Mysore, S. P., and Quartz, S. R. (2007). Why let networks grow. In Mareschal, D., Sirois, S., Westermann, G., and Johnson, M. H., editors, *Neuroconstructivism: Perspectives and prospects*, volume 2, chapter 4, pages 65–98. Oxford: Oxford University Press.

[Szirtes et al., 2005] Szirtes, Gábor, Póczos, Barnabás, and Lőrincz, András (2005). Neural Kalman filter. *Neurocomputing*, 65-66:349–355.

[Todorov and Jordan, 2002] Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235.

[Weng et al., 2001] Weng, Juyang, McClelland, James, Pentland, Alex, Sporns, Olaf, Stockman, Ida, Sur, Mriganka, and Thelen, Esther (2001). Autonomous mental development by robots and animals. *Science*, 291(5504):599–600.

[Westermann and Mareschal, 2004] Westermann, G. and Mareschal, D. (2004). From parts to wholes: Mechanisms of development in infant visual object processing. *Infancy*, 5(2):131–151.

Dr. Qinggang Meng is a lecturer in Department of Computer Science/Research School of Informatics, Loughborough University, UK. He is working in the areas of biologically and psychologically inspired learning algorithms, adaptive behaviour, service robotics, human-robot interaction, human motion analysis, driver cognitive distraction and computer vision. Before he joined Loughborough University, Dr Meng worked as a postdoctoral research associate for 4 years in the University of Wales, Aberystwyth(UWA), UK. He obtained his PhD from the Department of Computer Science, UWA. Before He came to UK, he did one year RA in City University of Hong Kong on a project of intelligent control of multifingered robot hand and its dynamic stability analysis. Before that, he worked in Intelligent Machine Institute at Tianjin University for several years in the area of intelligent robotics and image processing.

Mark Lee is currently Professor of Intelligent Systems in the Department of Computer Science, at the University of Wales, Aberystwyth, UK. He has been founder and Director of the Centre for Advanced Software and Intelligent Systems and a co-founder Director of the European Network: MONET.

His current research interests concern new methods of learning for robots. In particular, he is interested in why and how human infants can achieve so much learning - and robots so little. He believes the human brain can teach us much in this area but, surprisingly, the advances made in understanding human behaviour have not been given much attention until recently. His other research interests include Model-Based Reasoning with applications in engineering. He has developed qualitative electrical circuit models and am currently involved in mechanical and chemical processing plant modelling.

Dr Chris Hinde is a senior lecturer in Department of Computer Science/Research School of Informatics, Loughborough University, UK. He is the leader of the Intelligent and Interactive Research Group. He works mainly in various area of computational intelligence, including fuzzy, evolutionary and neural systems. His current interests lie in the area of fuzzy reasoning and evolutionary systems, in particular dealing with and reasoning about inconsistent and contradictory evidence. He believes that dealing with contradictory evidence will make useful contribution to machine learning as the presence of a contradiction in perceived evidence is an indication of a structural fault in the underlying knowledge framework.