International Journal of Automation and Computing 10(5), October 2013, 414-424
DOI: 10.1007/s11633-013-0738-5

Learning Robotic Hand-eye Coordination Through a Developmental Constraint Driven Approach

Fei Chao Xin Zhang Hai-Xiong Lin Chang-Le Zhou Min Jiang

Cognitive Science Department, Fujian Provincial Key Laboratory of Brain-like Intelligent Systems, Xiamen University,
Xiamen 361005, China

Abstract: The skill of robotic hand-eye coordination not only helps robots to deal with real time environment, but also affects the fundamental framework of robotic cognition. A number of approaches have been developed in the literature for construction of the robotic hand-eye coordination. However, several important features within infant developmental procedure have not been introduced into such approaches. This paper proposes a new method for robotic hand-eye coordination by imitating the developmental progress of human infants. The work employs a brain-like neural network system inspired by infant brain structure to learn hand-eye coordination, and adopts a developmental mechanism from psychology to drive the robot. The entire learning procedure is driven by developmental constraint: The robot starts to act under fully constrained conditions, when the robot learning system becomes stable, a new constraint is assigned to the robot. After that, the robot needs to act with this new condition again. When all the contained conditions have been overcome, the robot is able to obtain hand-eye coordination ability. The work is supported by experimental evaluation, which shows that the new approach is able to drive the robot to learn autonomously, and make the robot also exhibit developmental progress similar to human infants.

Keywords: Robotics, developmental robotics, developmental learning, robotic hand-eye coordination, robotic reaching.

1 Introduction

Robots supported by artificial intelligent algorithms have been able to execute highly complicated missions. However, if the robots are placed in completely unknown working environments, the robots often find itself difficult to perform properly. This is because the internal representations of the robot are foreknown and designed for expected environments[1]. It is a crucial issue that robots should be able to create their own internal representations. This problem has been solved in cognitive biological systems through processes of structured growth known as "development". Robotic scientists apply this idea to produce a novel research topic "developmental robotics". In the field of developmental robotics, it is assumed that a robot system is not programmed for specific and fixed tasks. The robot is programmed to develop and learn new behavioral and cognitive competence and skills autonomously[2]. This also means that a robot should be able to create its own internal representations through developmental learning. Developmental learning is concerned with not just the ability to learn and gain mastery at a given task but more importantly with how learning may progress and grow to achieve competence over a series of new tasks as they are encountered[3].

Developmental robotic draws inspiration from various aspects of developmental psychology and developmental neuroscience, e.g., sensory-motor coordinations, emergent behaviors, and social interactions[4]. In particular, very early development such as growth of sensory-motor control and skills plays an important role in both human infant and robotic cognition. Because early experiences and structures are likely to underpin all subsequent growth in ways that may be crucial. This agrees with the suggestion that sensory-motor coordination is likely to be a significant general principle of cognition[4]. Robotic hand-eye coordination belongs to the sensory-motor coordination. In this case, this paper emphasizes that it is very important to design learning algorithms of a robotic hand-eye system to achieve mastery of its local, egocentric space and to perform reaching task. This is because if we can understand the implementation of low level sensory motor, we might also apply those guidelines to build complex cognitive tasks in robotics.

From the viewpoint of developmental robotics, features from human development are required to be involved within robot training schema. In terms of this perspective, a number of models on robotic reaching and robotic hand-eye coordination have been proposed recently. Various types of constructive neural networks were proposed to build the mapping systems[5−7], in which visual perceptions are transformed into hand motor values. In other studies, incremental learning models were applied to learn such kind of mapping[8, 9]. Several other developmental robotic hand-eye coordination systems used different neural networks to simulate a part of brain working loops[10−12]. Such research indicates that introducing brain inspired structures into developmental robotics is regarded as an effective solution to robotic cognition[2]. However, compared with psychological theories and results, the aforementioned work merely adopted a few pieces of development features. In fact, there exists a large gap between our psychological theories of development and our ability to implement working developmental algorithms in autonomous agents.

In order to improve the current implementations, an integration of both a brain inspired computational structure and a significant developmental learning algorithm is appli-

Manuscript received February 22, 2013; revised July 4, 2013
This work was supported by National Natural Science Foundation of China (No. 61203336, 61273338 and 61003014) and Major State Basic Research Development Program of China (973 Program) (No. 2013CB329502).

ed to guide an autonomous robotic system to learn its hand-eye coordination. Adopting a brain inspired structure is realized as a key fact by reviewing aforementioned researches. By mimicking the mechanisms of the brain system, we would be able to learn important design principles for autonomous robotic systems, and bring higher autonomy into developmental robotics. Furthermore, implementing a brain inspired structure may support our robot to approach the complexity of cognition and adaptive behavior that we associate with biological organisms and may find their way into practical applications. This work, therefore, proposes a novel computational structure with two neural networks to map visual space to the robotic hand motor system. On the other hand, we believe that features of infant development can be used to guide our robot to develop. Therefore, "lifting constraints, act and saturate (LCAS)" algorithm is introduced to control the robot to gradually gain its hand-eye coordination ability. The algorithm works by removing several types of developmental constraints, which are summarized by psychological literatures[13]. LCAS algorithm was applied in several developmental models successfully[14−18]. This approach, combining the brain inspired structure and LCAS algorithm, will considerably widen the scope of psychological inspirations for developmental robotics, deepen the research on robotic reaching and hand-eye coordination, and bring more psychological and biological ideas into the developmental robotics.

The rest of this paper is structured as follows. Section 2 reviews the background and related work of robotic hand-eye coordination, reaching and developmental learning. Section 3 describes the computational brain inspired learning system of the robotic system and the development constraint driven approach in detail. Section 4 gives the experimental procedure and results. The discussions of those results and the comparison with other robotic hand-eye coordination models are mentioned in Section 5. Section 6 concludes the paper and points out important future research.

2 Background and related work

The core technique of the robotic reaching ability is to implement the hand-eye coordination, which is the mapping from robotic visual sensors to robotic actuators[17]. Mostly, the traditional robotic hand-eye coordination systems are calibrated by human engineers. However, this paper will not build this kind of work again, because this traditional method can work very precisely[19−21]. The paper emphasises that robots themselves are able to learn the mapping by self-calibration.

Fig. 1 shows a diagram of the typical robotic hand-eye mapping problem. The system obtains the location of a target within an image captured by a visual sensor, this information is then converted to the eye-centered coordinate system. After that, the eye-centered information is mapped to the hand space through a hand-eye transformation mechanism. Then, the difference between the current hand position and the desired position is used to drive the hand motors. This architecture of decomposing sensory-motor transformation into sub-transformations is supported by findings in human brains during the performance of visually guided reaching. In the architecture, the position of the target is re-mapped into various intermediate frames of reference, such as eye-centered and hand-centered. The hand-eye transformation problem is thus formulated as finding the relationship between the eye motor space and the arm joint space, i.e., find the mapping: $f : (m_1, m_2, \cdots, m_k) \rightarrow (j_1, j_2, \cdots, j_n)$, where m_i is the i-th eye motor value, j_l stands for l-th arm joint motor value, k is the number of motors to drive the eye movements, and n is the number of the arm joints.

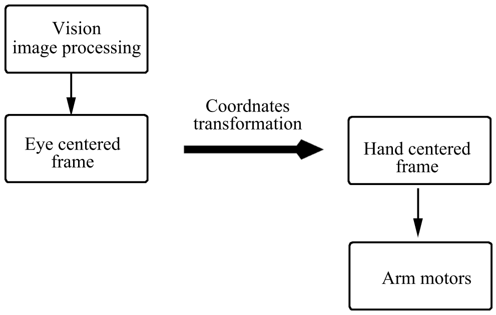


Fig. 1 A typical robotic hand-eye coordination diagram

Hand-eye coordination in developmental robotics is an important topic. Therefore, a number of robotic reaching or robotic hand-eye coordination models have been proposed recently. However, those works have limitations due to various facts: 1) Most of the robotic hand-eye coordination systems only use one camera as the robotic vision system, merely carry out 2 dimensional experiments which might simplify their systems learning complexity[8, 9, 22]. 2) The training phase ignores imitating the developmental progress of human infants reaching objects[23, 24]. Khamassi et al.[12] inspired us that we can build a computational model to simulate part of brain working loop. However, they only focus on building a complex system, rather than let a system develop to be complex. Nori et al.[25] attempted to make their robot autonomously build an "eye-to-hand" Jacobian matrix to implement their reaching ability. This approach is able to deal with the kinematic redundancy of the robot. Nevertheless, developmental behaviors are not shown in the research. Jamone et al.[26] designed two types of reaching maps to build the hand-eye coordination: One map estimated whether an object can be reached, and another map supplied enough information for reaching. However, this work is merely done in simulated environment. Our previous work is similar to [27, 28], where we applied neural networks to control our robot to learn reaching.

It is important to note that the hand-eye mapping is highly nonlinear. The nonlinear approximation ability of the artificial neural network supports the implementations of the hand-eye mapping very well. In particular, [10, 12] used radial basis function (RBF) networks to simulate V6A cortex in the human brain. Self-organizing map networks and Jacobian matrices are able to deal with the kinematic redundancy of the robot[11, 29]. Furthermore, scientists of developmental robotics focus on bringing more developmental features into their researches, therefore an enhanced model of minimal resource allocation network is proposed to build a mapping system[5, 6], the network topological structure grows while the network is being trained. The growing structure feature fits psychological findings properly. Zhou et al.[7] also modified a self-organization map network to gain the incremental structure feature. Meng and Lee[30] used active learning algorithm to drive their robot to use more training to learn the difficult subspace of their robot arm and use lesser training to learn the simple subspace.

Human infants are a successful learning model for hand-eye coordination. Therefore, in order to build the development-driven approach, it is necessary to understand human infant development procedure, and it is crucial to abstract the significant developmental features from the procedure. Therefore, the following subsections introduce the infant development model and developmental constraint, respectively.

2.1 Inspirations from infant development

Literatures from developmental psychology show the developmental procedure of human infants. The procedure demonstrates that the reaching movement starts from coarse reaching movements to precise[31, 32]. After birth, human infants have a few visuo-motor ability, and can perform directed hand movements toward visual targets. These "pre-reaching" movements are not successful in making contact with targets. However, infants perform visually directed actions to reach the "hemifield", in which a target appears. It is clear that early reaching is based on vision of the target and on proprioceptive signals from the arm. Later, during the last months of completing the age of one year, infants use the distance between the hand and the target to adjust the hand position and orientation, so as to touch target object[31, 33]. This type of movements is like corrections of failed reaching movements. Therefore, our robotic system can follow the infant′s pattern: A reaching movement consists of the pre-reaching movements and the correction movements.

On the other hand, from the aspect of neural processing modules, human infants′ reaching is the result of the cerebral cortex, basal ganglia, and cerebellum working in parallel to generate and control[34]. At the age of 15 weeks, the basal ganglia-cerebral cortex loops select a rough action that might need correction during execution. In adults, these actions would be refined through the action of the cerebellar loops. However, this type of action is limited in young infants because of their limited development of the cerebellum. With repeated experiences, the basal ganglia loops become better at selecting actions for particular contexts using reinforcement learning. As time passes during the first year and the cerebellar network becomes more mature, the cerebellar cortical loops should gradually exert influence on the approximate commands selected by the basal ganglia loops[32]. In terms of these findings, we consider the robot will contain two different control mechanisms to handle the two distinct movement patterns.

2.2 Developmental constraints and LCAS algorithm

The above findings clearly show the infant developmental procedure during 0–12 months after birth. However, it is also important to identify what is the internal force that drives infant to develop from behaving coarse reaching movements to accurate ones. A robotic developmental driving mechanism is required to mimic the similar developmental procedure, so our robotic system can use the driving mechanism to determine when to develop new abilities. Developmental psychology shows evidences that lifting constraint can lead infants to develop from one competence to a new, and even more complicated competence. This is because each developmental step establishes the boundary conditions for the next one. A particular ability cannot emerge if any of the capacities it entails is lacking[1]. In addition, our previous work[13] proposed the LCAS algorithm, which deals with discrete developmental stages, whose maturation level increases, depending on a notion of saturation linked to the estimation of novelty. Therefore, we apply the LCAS algorithm to implement a developmental mechanism: Let the robot act under a constraint. When a constraint has been saturated, a new constraint is released into the system. To further ease the understanding of LCAS, Algorithm 1 shows an outline of the constraint lifting procedure in pseudo code. The value of $sat(x, \Delta, n)$ indicates whether the learning system is stable.

Algorithm 1. LCAS algorithm

```
1) while not all constraints are released do
2)     for $i = 0$ to $n$ do
3)         if $sat(x, \Delta, n)$ then
4)             quit this for-loop and release a new constraint
5)         else
6)             repeat doing the learning process within this for-loop (from Line 2)–8))
7)         end if
8)     end for
9) end while
```

Inspired by the two types of movements and the two areas of brain cortices mentioned above, this work proposes to design a robotic learning system with two constructive neural networks. Upon this setup, one neural network is trained to control our robot′s large amplitude of arm movements around the objects. The network is to produce the early infant pre-reaching, and imitate basal ganglia-cerebral cortex loops performing rough actions. Another neural network is designed to behave in accordance with small amplitude arm movements to make correct reaching movements. Thus, the second network generates the later movement of infant reaching. Also, a developmental mechanism guides the robot when to develop from the first network to the second based on its learning status. Note that, our work simply simulates the basic functions of the basal ganglia loops, rather than simulates every function in detail.

3 Methods

The experiment aims to achieve the goal that the robotic arm can learn to capture target objects behavior through the robot′s spontaneous movements. This section describes the two neural networks architecture, the learning algorithms for training the robotic learning system, and the robotic system configuration.

3.1 Brain inspired network structure

The entire robotic learning system proposed in this paper consists of two neural networks. The first network N_1 simulating the basal ganglia cortex is able to generate rough reaching movements, and the second network N_2 simulating the cerebellum can make correction movements after a rough reaching has been made. These two networks are not used in parallel. We set up a threshold δ to choose which network to be used. Hence, we calculate the distance between the target position and the hand position within the images captured by the two cameras. If the distance is greater or equal to the threshold, N_1 is active; else, N_2 is active. Because N_2 can give more accurate motor commands to the arm, the structures of N_1 and N_2 are not identical. In the training phase, there is no target object appearing within the robotic vision′s view, the robot can only detect the position of the fingertip. The whole structure can be divided into three sub modules: the image processing module, the hand sensory-motor module, and the network control module. The following sections describe this brain inspired network structure and those sub-modules in detail.

3.1.1 Image capture and image processing modules

In order to detect the fingertip in the images captured by the two cameras, we first need to convert the image from the RGB color space to the HSV color space. Then we use the foreground generation method to generate the histogram of finger. In the system, if Hue value of a pixel falls into the range: Hue $[245, 253]$, we set this pixel as a candidate target pixel. Then we assign 255 to this pixel; otherwise, we assign 0. An orange ball is used as the target to be captured, its color range is Hue $[29, 42]$. After the color conversion, a binary image is generated. The candidate target pixels are further grouped into several regions, and the largest region is regarded as the fingertip against the background of our experimental system. The center of the region is used as the fingertip position (x_a, y_a) of each image. The image processing result is shown in Fig. 7. In addition, if the object is very close to the fingertip, the arm can touch it without more movements, we consider this as a touchable state.

3.1.2 Hand sensory-motor module

This module can handle the movement of each joint and feedback its position. θ_1, θ_2, and θ_3 stand for the positions of the three motors, respectively. Each joint has its working range: θ_1 is $[0°, 60°]$, θ_2 is $[30°, 120°]$, and θ_3 is $[40°, 110°]$. This module only receives relative movement values $(\Delta\theta_{1-3})$ and converts them to the corresponding motor values that are sent to the hardware controller.

3.1.3 Network control module

When robotic arm has owned the reaching ability, a target object is placed within the arm workspace. The robotic vision system can sense the target position (x_{o1}, y_{o1}) in Camera 1, and (x_{o2}, y_{o2}) in Camera 2, and similar for the fingertip position (x_{a1}, y_{a1}) and (x_{a2}, y_{a2}). d is the Euclidean distance between the object and the fingertip. If d is greater or equal to δ, N_1 is applied to generate $\Delta\theta_{1-3}$ for the motors. Otherwise, the target position and the difference between the target and the fingertip $(\Delta x_{o-a1}, \Delta y_{o-a1}, \Delta x_{o-a2}, \Delta y_{o-a2})$ are sent to N_2 to generate $\Delta\theta_{1-3}$. Then, $\Delta\theta_{1-3}$ are sent to the motors. The procedure is illustrated in Fig. 2.

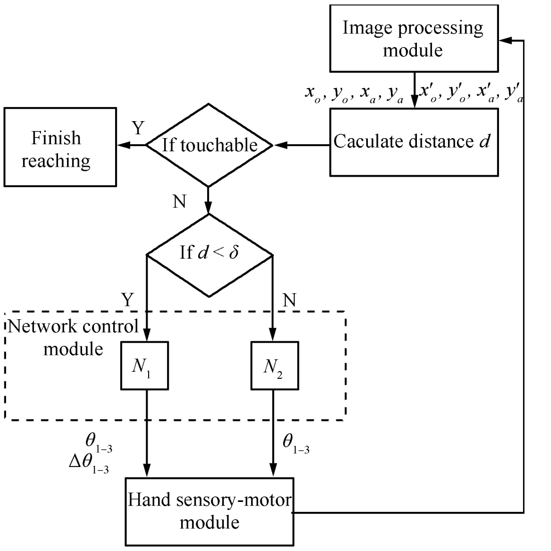


Fig. 2 The control procedure of the robotic system

3.2 Network training phase

The robot needs to generate a number of spontaneous movements to learn to handle its arm. The spontaneous movements mean the arm moves randomly. Before each movement, both the cameras calculate the fingertip position, (x_1, y_1) indicates the fingertip position in Camera 1, (x_2, y_2) gives the position in Camera 2. Meanwhile, the joint values $(\theta_1, \theta_2, \theta_3)$ of the robotic motors are acquired from the hand sensory-motor module. After one movement, both the fingertip position and the joint values are changed. We use $(x'_1, y'_1), (x'_2, y'_2)$ and $\theta'_1, \theta'_2, \theta'_3$ to identify the new values. Note that the Euclidean distance d between $(x_1, y_1), (x_2, y_2)$ and $(x'_1, y'_1), (x'_2, y'_2)$. d can be calculated by (1). $\Delta\theta_{1-3}$ are the different values between θ_{1-3} and θ'_{1-3}.

$$d = \sqrt{\sum_{n=1}^{2}(x_n - x'_n)^2 + (y_n - y'_n)^2} \tag{1}$$

where (x_n, y_n) are the hand position within the images captured by Camera n before a movement, and (x'_n, y'_n) mean the hand position after the movement.

The threshold δ can be used to determine which network is trained or used to control the arm. In the experiment, δ is set to 16. If d is larger than or equal to δ, N_1 network is selected. This is because if d is large, the movement scope of the arm is large, only N_1 can handle this large movement in the architecture. $(x'_1, y'_1), (x'_2, y'_2)$ are the inputs of N_1, and $\Delta\theta'_{1-3}$ are the network′s expected outputs. N_1 is trained by using $(x'_1, y'_1), (x'_2, y'_2)$ and θ'_{1-3} as the training pattern. If d is less than δ, N_2 is trained, its inputs contain $(x_1, y_1), (x_2, y_2)$ and $(\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2)$, its expected outputs have $\Delta\theta_{1-3}$. The training procedure is shown in Fig. 3.

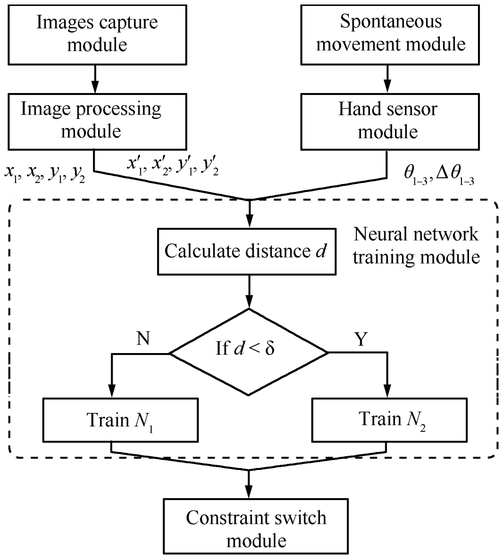


Fig. 3 The training schema of a brain inspired network architecture

3.2.1 Constructive neural network implementation

We use a type of constructive neural network to build the robot learning system, because psychologists indicate that constructive learning occurs not only in infancy but also in mature brains[35]. The constructive learning means learning system can grow up automatically during learning phase. The network used in this paper is called minimal resource allocating network (MRAN). MRAN network starts with no hidden units, and with each learning step, i.e., after an action, the network grows or shrinks when necessary or adjusts the network parameters accordingly.

A typical MRAN network is expressed as

$$f(x) = \alpha_0 + \sum_{k=1}^{N} \alpha_k \phi_{k(x)} \tag{2}$$

$$\phi_{k(x)} = \mathrm{e}^{-\frac{1}{\sigma_k^2}\|x-\mu_k\|^2} \tag{3}$$

where α_k is the weight vector from the hidden unit $\phi_k(x)$, N is the number of radial basis function units (N of N_1 is 15, N of N_2 is 30), and μ_k and σ_k are the k-th hidden unit′s center and width, respectively. $f(x) = (f_1(x), f_2(x), \cdots, f_{N_O}(x))^{\mathrm{T}}$ is the network output vectors, where N_O is the number of the MRAN network outputs, and x is the network input. In the brain inspired structure, N_1 network setup is addressed as follows: $f(x) = \{\Delta\theta_1, \Delta\theta_2, \Delta\theta_3\}^{\mathrm{T}}$ is the vector of the arm joint angles, and $x = \{x_1, y_1, x_2, y_2\}^{\mathrm{T}}$ is the vector of the target positions within both Camera 1 and Camera 2. N_2 network configuration is $f(x) = \{\Delta\theta_1, \Delta\theta_2, \Delta\theta_3\}^{\mathrm{T}}$, and $x = \{\theta_1, \theta_2, \theta_3, \Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2\}^{\mathrm{T}}$.

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 larger than a threshold, and whether the node to be added is far enough from the existing nodes in the network, as shown in (4) and (6). The criterion in (5) is to check the prediction error within a sliding window to ensure that growth is smooth, m is the length of the sliding window.

$$\|e(t)\| = \|y(t) - f(x(t))\| > e_1 \tag{4}$$

$$\sqrt{\sum_{j=t-(m-1)}^{t} \frac{\|e(j)\|^2}{m}} > e_2 \tag{5}$$

$$\|x(t) - \mu_r(t)\| > e_3 \tag{6}$$

where $x(t), y(t)$ are the t-th learning data, m is the sliding window size, and $\mu_r(t)$ is the centre vector of the nearest node to $x(t)$, e_1, e_2 and e_3 are the three thresholds. If the above three conditions are not met, the extended Kalman filter is applied to modify the weights of the network, so as to decrease the network error, and a node is inserted to the network. In the experimentation, $e_1 = 0.05, e_2 = 0.005$, $e_3 = \max\{0.4 \times 0.999^i, 0.07\}$, and i is the learning step.

3.2.2 Spontaneous movement module

The spontaneous movement module provides random motor values to the robot system. This setup is to follow the feature that human infants always apply this type of spontaneous movements to build their hand-eye coordination ability. The following equation is used to implement this module.

$$M_i = \mathrm{rand}((M^i_{\mathrm{MAX}} - M^i_{\min}) + SR \tag{7}$$

where i is the i-th joint, $M_{\min}$ is the position which cannot be small, M_{MAX} is the position which cannot be large, and SR is a safety parameter, which is to drive the arm not to move to the largest or the smallest position. Otherwise, moving into those positions may damage the motors. M_{MIN} is the constraint applied in this paper, which has a larger value when the system starts to act, and changes to a smaller value afterwards.

$$M^i_{\min} = M^i_{\mathrm{MIN}} \times \mathrm{e}^{-\beta} \tag{8}$$

where β is a parameter to control the decreasing speed of the arm′s moving range, and $\beta \in \{0.6, 1.0\}$.

3.3 Developmental mechanism

In this paper, only one constraint is raised or relaxed to drive the development of the whole robotic learning system. The constraint is the movement amplitude of robotic arm. There are five types of developmental constraints: 1) Anatomical, 2) sensory-motor, 3) cognitive, 4) maturational, and 5) external constraints[13]. The movement amplitude belongs to the sensory-motor type. The constraint-lifting procedure can be described as follows: At the beginning, the arm can only wave with large amplitude, the robotic system uses these rough movements to train the network N_1. As the change of the constraint, the robotic arm starts to train the other network N_2 to refine reaching movements until the arm could make correct reaching movements. Then, the whole system becomes stable and mature.

3.3.1 LCAS algorithm with the brain inspired architecture

Our approach of shaping similarly discovers the structure of competence possibilities under a given constraint regime. This consists of implementing the cycle, lift-constraint, act and saturate (LCAS). First, we identify all possible and available constraints and decide which should be initially applied to the system. Next, we execute the motor action algorithm described below. This uses motor babbling to discover any irregularities in the sensory-motor modalities and stores these in explicit schemas. Eventually, the motor babbling will not produce any new space, as indicated by very low global excitation, and then we consider the system to be saturated. At this point, a constraint can be eased or lifted and the cycle starts again. The constraints can be scheduled according to infant maturational data, but we have also experimented with automatic schemes where the selection of constraints varies as an emergent process.

In terms of the above considerations, our developmental mechanism is implemented to drive the constraints to change. As the times of arm movements increase, the system tends to become more saturated. When the saturation value is stabilized and is less than a fixed value ψ, we could regard the robotic system as saturated. Therefore, we increase the robotic movement range to acquire new training data to retrain the robotic learning system, until the saturation value of the whole system is stabilized and the value is less than the fixed value ψ. The mechanism can be summarized as the following equation:

for $t = \varphi, \cdots, n$

$$sat(y(t), t) = \begin{cases} \text{true}, & \text{if } |y(t) - y(t-\varphi)| < \varepsilon \\ & \text{and } y(t) < \psi \\ \text{false}, & \text{otherwise} \end{cases} \tag{9}$$

where in the experiment, φ, ψ and ε are set to 0.1, 0.02 and 5, respectively.

Equation (9) gives the saturation rule: t means the number of training epoches, $y(t)$ means the global excitation value of the system at epoch t, ψ is the fixed value. If $y(t)$ is less than φ for a certain period of time φ, we will regard this as the saturated situation, the constraint then lifts. In short, (9) means if the learning system remains stable for a fixed term, a new constraint is assigned into the system.

It is very important to define $y(t)$ carefully. The network output error values are not used in $y(t)$ directly. Instead, we apply a habitation equation to define $y(t)$. At each stage of learning, novelty and habituation play an important role in driving the learning process. Novelty refers to new or particularly salient sensory space, while habituation is defined as a decrease in the strength of a behavioral response to repeated stimulations. A habituated stimulus may be able to evoke a further response after the presentation of an intervening novel stimulus. Novelty and habituation mechanisms can help a system to explore new places/events while monitoring the current status. Therefore, the system can glean experience over its entire environment. In our system, we used a biologically plausible habituation model which was created in our previous work[36, 37] describing how excitation, y, varies with time

$$y(t+1) = y(t) + \frac{\alpha}{\tau}[y_0 - y(t)] - \frac{S}{\tau} \tag{10}$$

where y_0 is the original value of y, τ and α are time constants governing the rate of habituation and recovery. In the experiment, we set τ, α and y_0 to be 5, 0.9 and 1.1, respectively. S indicates the system is in habituation or the novelty model. If the network output error oe keeps decreasing in the training process, we regard the system to be in the habituation model $S = 0$. Otherwise, if oe suddenly increases during the decreasing trend, we consider the increasing as a novel stimulus, thus, $S = 1$. If $sat(y(t), t)$ in (9) is true, the learning process of N_2 starts, but N_1 stops. If $sat(y(t), t)$ is false, the training still occurs in N_1.

3.4 Training method

We also set a short-term memory system to hold training data, which are obtained from the robot learning phase. After each hand movement, the visual data and the hand joints data are inserted to the memory. Then, the robot uses the data in the memory to train N_1 or N_2. The network will keep training itself until it gets convergence. Note that the memory format of N_1 is different from that of N_2, since the two networks contain diverse inputs. The memory has a capacity: The oldest data will be discarded when new data are inserted.

4 Experimentation

In this section, the results of a number of experiments are reported to demonstrate the capabilities of the proposed approach.

4.1 Hardware

Fig. 4 illustrates the experimental robotic system, which is an "InnoStar" robotic arm including 6 degrees of freedom (DOF). The arm is mounted on a workspace, 3 DOF of the arm are used in this paper to finish reaching movements. There is a gripper with two fingers mounted on the top of the arm. This setup can support the robotic arm to move and capture objects in 3-dimensional environment. Each rotational joint of the robot arms has a motor driver and also an encoder which senses the joint angle, thus providing a proprioceptive sensor reading. The upper limb and the lower limb are labeled as L_1 and L_2, respectively. The length of L_1 is 14 mm, L_2 is 20 mm.

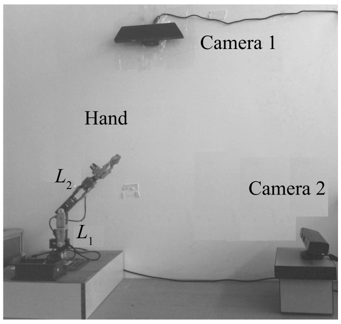


Fig. 4 The experimental hardware

Two RGB cameras are applied to build the robotic vision system in this work. Because one camera cannot supply enough information for 3-dimensional reaching movements, a camera ("Camera 1" in Fig. 4) is mounted on a frame placed next to the arm, and another camera ("Camera 2" in Fig. 4) is mounted above and looks down on the work space. The robotic fingertip and the object are marked with different colors so as to be detected easily.

Fig. 5 demonstrates the robotic control system which is divided into two parts: An AVR controller and a host computer. Both motor and sensor systems controllers are installed on the AVR controller. A program including RSC-232 socket and integrating the controllers driver programs have been built for communications between the host computer and the AVR controller. The program also simplifies the command language of the controller to remotely invoked functions so that the controllers can be called by other on-line host computers on conveniently. Therefore, any computer running the developmental algorithms can control the robot arm via the AVR computer. In fact, the AVR computer is like an interface between the laboratory robot and the outside world. Therefore, the high level application program on the host computer will never consider the underlying controllers drivers, but merely concentrate on the developmental learning algorithms.

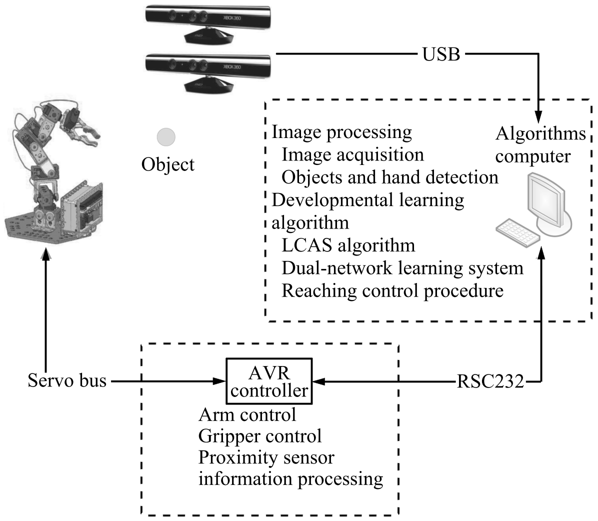

USB
Image processing
Image acquisition
Objects and hand detection
Developmental learning algorithm
LCAS algorithm
Dual-network learning system
Reaching control procedure
Algorithms computer
Object
Servo bus
AVR controller
Arm control
Gripper control
Proximity sensor information processing
RSC232

Fig. 5 The computer control system

Fig. 6 shows the output of image processing module. Figs. 6 (a) and (c) are the images captured by Camera 1 and Camera 2, respectively. Fig. 6 (c) highlights the fingertip from Fig. 6 (a), and the fingertip within Fig. 6 (c) is also highlighted in Fig. 6 (d). The image processing module can detect the fingertip positions within both Figs. 6 (b) and (d).

4.2 Experimental results

The experimental procedure can be designed as follows: Mark the hand of the robotic arm and the target by using a particular color. Firstly, no object is put into the workspace. The learning system only generates spontaneous movements so that the learning patterns can be generated by capturing and calculating these movements. According to the saturation of the brain inspired learning system, the robot arm hardly performs small range of movements at the beginning, but will change to generate more small range of movements in the middle term of the experiment. After the learning phase has been completed, a static object is put into the workspace, and then, the robot attempts to touch the object.

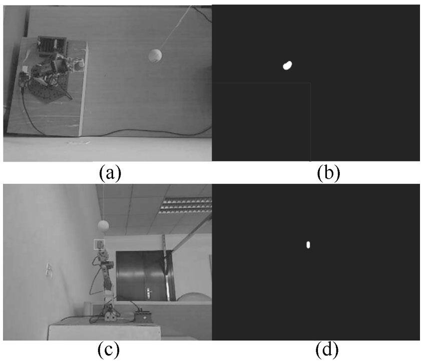


Fig. 6 The output of the image processing module

Fig. 7 demonstrates the movement variation of the entire learning phase: The curve labeled "Movements for N_1" shows how many movements are used to train N_1 network during the experiment, and the curve labeled "Movements for N_2" stands for the number of the random movements used to train N_2 network. Because the small range of movements are ignored during the beginning period of the experiment, N_2 network has no chance to get trained. Only the large range of movements are accepted by N_1 network. After about 90 movements, according to (9), the robot begins to have several small range of movements by using (7) and (8), N_2 network starts to learn those movements. Note that the large range of movements are still generated by the robot after about the 90th movement. However, those movements are not used to train N_1 any more. From this figure, the robotic behavior pattern is to learn coarse movements first and learn precise movements later. This setup successfully simulates infant developmental feature, which is infant′s movement starting from coarse to precise ones.

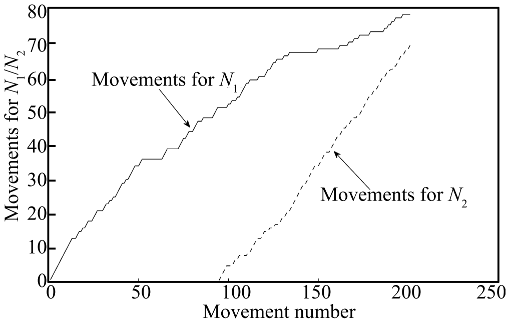


Fig. 7 The movement types during the learning procedure

Fig. 8 shows that the saturation of the robot system during the overall training phase. The global excitation value, which is calculated by (10), is used to indicate whether the learning system is stable. The curve before the "N_2 starts to work" label is generated by N_1 network. The global excitation value keeps oscillating until after about 100 movements, and then it falls down. When the global excitation value satisfies (9), the curve mainly reflects N_2 network saturation, because the training only occurs in N_2 network. The rest curve returns to oscillate, and it becomes stable again at around 280th training. At this moment, the robotic learning is completed, and the robot has owned the ability to make reaching movements. All these situations are caused by the changing of the constraint that we set in the robotic system. Therefore, the developmental procedure of the entire robotic system is driven by the constraint. In addition, Fig. 8 also indicates that the convergence speeds for both N_1 and N_2 are fast.

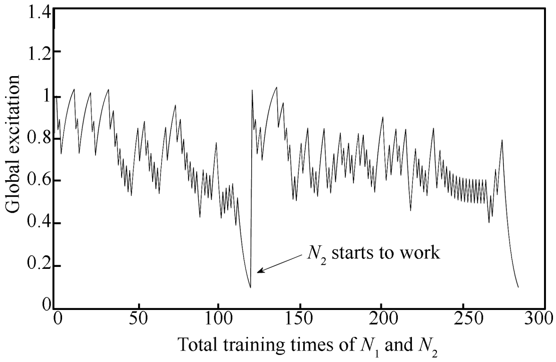


Fig. 8 The saturation during the training phase

The two network output errors within the entire training procedure are shown in Figs. 9 and 10, respectively. N_1 network requires about 120 trainings to achieve convergence, and N_2 network needs nearly 170 times to converge. This difference indicates that the network system is able to learn the large range of movements more easily than to learn the small range of movements. Furthermore, Figs. 9 and 10 also imply that development constraint divides one task to several sub-tasks. The robot completes each sub-task one by one, and the robot starts to learn the easier part first. Thus, difficulty of the entire task is reduced. Note that the results of network output errors are for the entire workspace, rather than a fixed position. Therefore, after the robot completes its training, no matter where the object is placed, the robot is able to generate precise reaching behaviors.

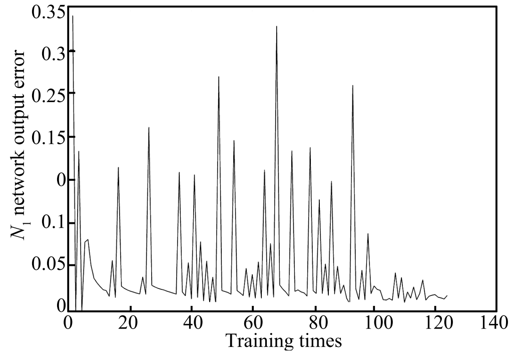


Fig. 9 N_1 network output error

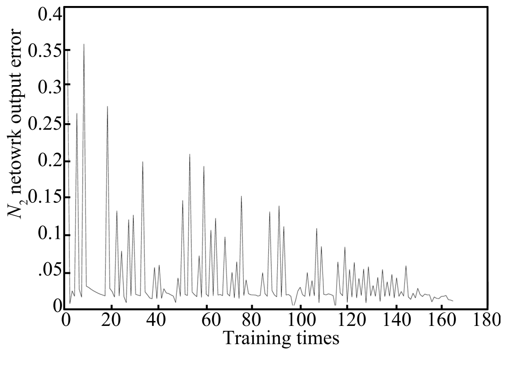


Fig. 10 N_2 network output error

Fig. 11 shows the arm positions before and after a reaching movement. An orange pingpong ball used as a target was hung into the robot workspace, then the arm prepares to move from a position (which is far from the target) towards the target. The left picture shows the start position, and the right picture shows that the target is already within the robotic gripper. Thus, Fig. 11 illustrates that our robot has been able to exhibit successful reaching movements. However, the movement details are not shown in the figures. In this case, Figs. 12 and 13 demonstrate the movement trajectories.

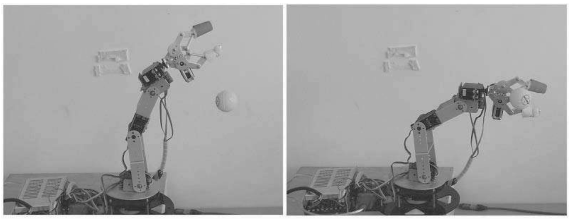


Fig. 11 A reaching movement

Both Figs. 12 and 13 show the robotic arm trajectories of a successful reaching movement. The x axes of Figs. 12 and 13 are labeled by the horizontal pixel number of the images capture by Cameras 1 and 2, and the y axes are the vertical pixel number of the images. Fig. 12 gives the fingertip trajectories after each movement from Camera 1, and Fig. 13 shows the trajectories from Camera 2. This successful reaching contains 3 movements which are drawn as 3 arrows and labeled as “1”, “2” and “3” according to the movement sequence. We can obverse that the range of the first movement is very large. However, this movement cannot reach the object. Then, two correct movements lead the hand touch the object. Compared to the first movement, the second and the third movements are relatively short.

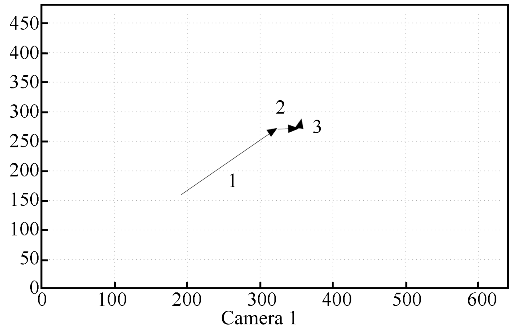


Fig. 12 The trajectories of the arm taken by Camera 1

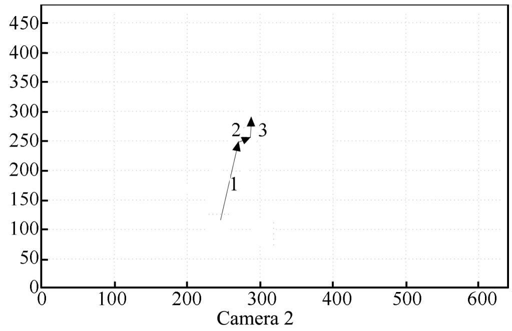


Fig. 13 The trajectories of the arm taken by Camera 2

In this case, the robotic system has to apply two different types of movements to finish one reaching movement. Thus, this movement pattern is, similar to human infants, or even adult′s reaching movement pattern.

5 Discussions

The experimental results described above have demonstrated how the robot learns to map from its visual stimuli to its hand, and how the brain inspired network structure and the developmental learning mechanism cooperate together to drive the learning process. In order to specify the advantages of our approach, a comparison with other developmental approaches is given in Table 1. We have not compared our approach with every work mentioned in Section 2, we only summarize features of those work. Based on this point of view, the following features are compared: 1) The hand-eye transformation method, 2) the biological plausibility, 3) the developmental stages, 4) the incremental learning, the learning speed, and 5) the human-like movement pattern.

In Table 1, we find many existing approaches prefer to apply the static neural networks to implement the eye-to-hand nonlinear transformation. However, our approach is to handle this transformation by using the constructive neural network. This type of neural network contains an important feature: The network topology and weights are changed simultaneously during its training phase. The feature fits several important psychological theories on cognition[4]. Upon the biological plausibility factor, the regular methods merely focus on the transformation implementation. Only a few works report that they simulate some brain cortices functions that are used to guide human reaching movement. However, our approach is inspired by the control loop of the cerebellum and the basal ganglia in human brain. Therefore, our method has more biological plausibility. And those existing works did not show much how their competencies emerged gradually. They built every function together to enable the systems to develop. By contrast, our approach can make the robotic system built up from simple tasks to highly complex tasks, which makes the system functions became more and more proficient during its development. In particular, the experiment demonstrates a situation of staged behavior change, this is very similar to human infant developmental process. Another significant advantage is that both neural network topological structure and the learning system are incremental, other than some existing work did not own incremental learning. Also, because of the two neural networks structure, the overall learning time of the existing approaches and our work are totally different. Some works connectionist implementations that adopted single static neural network require thousands of cycles of training epoches. However, our system, consequently, uses less than 400 trials to achieve a matured level (see the two curves in Figs. 10 and 11). Finally, the reaching movement of the existing approaches is usually achieved by a single movement, but our approach is able to use a combination of coarse movements and correction movements to finish a reaching movement.

6 Conclusions

This paper extends the recent work on robotic hand-eye coordination and proposes a novel robotic learning system, which can drive our robot to gradually gain reaching movement ability. The method worked by first constructing a brain inspired computational structure to simulate human brain work loops via implementing two constructive neural networks; and then creating a developmental mechanism implemented by "lifting constraints, act and saturate algorithm" to drive the robot gradually and autonomously learn reaching ability. The observations from the experiments display an increasing progression from initially behaving large range of hand movements, and then to generating small range of movements. These indicate this approach not only improves the current work which brings ideas of developmental psychology and neuroscience into robotics, but also has other three advantages: 1) Our robotic learning progression is very similar to human infant′s development. 2) The approach incorporates incremental and cumulative features in its learning. 3) It owns more autonomous and psychological characteristics.

There is still room to improve the present work. In particular, the present work only uses two static cameras as the robotic vision, and merely carries out experiments within a settled workspace. But practically, variable environments for the working of robots may be more useful. Thus, we propose to use a motorized 2 DOF stereo vision to replace the two static cameras in the system, which can increase our robotic ability so as to work within more complicated environment. Finally, this work does not look into the threshold configuration. In the experiments, some thresholds' values are set manually. Further effort to investigating these issues seem useful.

Table 1 Summary of the comparison with the existing approaches

Option	Existing approaches	Our approach
Hand-eye transformation method	Artificial neural networks, and Jacobian matrix	Constructive networks
Biological plausibility	Only few features reported[10]	Basal ganglia-cerebral were simulated by dual networks
Developmental stages	Few works supported	Staged behavioral patterns and staged learning system patterns
Incremental learning	Few works supported	Both neural network topological structure and the learning system are incremental
Learning speed	Slow in several static neural network based approaches	Faster than single network approach
Human-like movement pattern	Directed movements	A successful movement contains a coarse movement and movement and several corrections

Acknowledgement

This paper was presented in part at the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO 2012). The authors would like to thank the anonymous reviewers who were very helpful in revising this paper.

References

[1] M. Lungarella, G. Metta, R. Pfeifer, G. Sandini. Developmental robotics: a survey. *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.

[2] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, C. Yoshida. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

[3] M. H. Lee, Q. Meng, F. Chao. Developmental learning for autonomous robots. *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 750–759, 2007.

[4] A. Stoytchev. Some basic principles of developmental robotics. *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 122–130, 2009.

[5] Q. Meng, M. H. Lee. Automated cross-modal mapping in robotic eye/hand systems using plastic radial basis function networks. *Connection Science*, vol. 19, no. 1, pp. 25–52, 2007.

[6] Q. Meng, M. H. Lee, C. J. Hinde. Robot competence development by constructive learning. *Advances in Machine Learning and Data Analysis, Lecture Notes in Electrical Engineering*, vol. 48, pp. 15–26, 2010.

[7] T. Zhou, P. Dudek, B. E. Shi. Self-organizing neural population coding for improving robotic visuomotor coordination. In *Proceedings of International Joint Conference on Neural Networks*, San Jose, California, USA, pp. 1437–1444, 2011.

[8] M. Huelse, S. McBride, J. Law, M. H. Lee. Integration of active vision and reaching from a developmental robotics perspective. *IEEE Transactions on Autonomous and Mental Development*, vol. 2, no. 4, pp. 355–367, 2010.

[9] M. Huelse, S. McBride, M. H. Lee. Developmental robotics architecture for active vision and reaching. In *Proceedings of IEEE International Conference on Development and Learning*, Frankfurt am Main, Germany, pp. 1–6, 2011.

[10] E. Chinellato, M. Antonelli, B. J. Grzyb, A. P. Pobil. Implicit sensorimotor mapping of the peripersonal space by gazing and reaching. *IEEE Transactions on Autonomous Mental Development*, vol. 3, pp. 43–53, 2011.

[11] P. P. Kumar, L. Behera. Visual servoing of redundant manipulator with Jacobian matrix estimation using self-organizing map. *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 978–990, 2010.

[12] M. Khamassi, S. Lallee, P. Enel, E. Procyk, P. Dominey. Robot cognitive control with a neurophysiologically inspired reinforcement learning model. *Frontiers in Neurorobotics*, vol. 5, no. 1, pp. 1–14, 2011.

[13] M. H. Lee, Q. Meng, F. Chao. Staged competence learning in developmental robotics. *Adaptive Behavior*, vol. 15, no. 3, pp. 241–255, 2007.

[14] J. Law, M. H. Lee, M. Huelse. Infant development sequences for shaping learning in humanoid robots. In *Proceedings of the 10th International Conference on Epigenetic Robotics*, Lund University Cognitive Studies, Sweden, pp. 65–72, 2010.

[15] J. Law, M. H. Lee, M. Huelse. The infant development timeline and its application to robot shaping. *Adaptive Behavior*, vol. 19, no. 5, pp. 335–358, 2011.

[16] J. Law, P. Shaw, M. H. Lee. A biologically constrained architecture for developmental learning of eye-head gaze control on a humanoid robot. *Autonomous Robots*, vol. 35, no. 1, pp. 77–92, 2013.

[17] F. Chao, M. H. Lee. An autonomous developmental learning approach for robotic eye-hand coordination. In *Proceedings of Artificial Intelligence and Applications*, Innsbruck, Austria, pp. 639-013-1–6, 2009.

[18] F. Chao, M. H. Lee, J. J. Lee. A developmental algorithm for ocularmotor coordination. *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 239–248, 2010.

[19] D. Xu, C. A. A. Calderon, J. Q. Gan, H. Hu, M. Tan. An analysis of the inverse kinematics for a 5-DOF manipulator. *International Journal of Automation and Computing*, vol. 2, no. 2, pp. 114–124, 2005.

[20] D. Xu, H. W. Wang, Y. F. Li, M. Tan. A new calibration method for an inertial and visual sensing system. *International Journal of Automation and Computing*, vol. 9, no. 3, pp. 299–305, 2012.

[21] H. B. Wang, M. Liu. Design of robotic visual servo control based on neural network and genetic algorithm. *International Journal of Automation and Computing*, vol. 9, no. 1, pp. 24–29, 2012.

[22] F. Chao, H. Lin, M. Jiang, M. Shi, J. Chao. Integration of brain-like computational structure and infant behaviorial pattern for robotic hand-eye coordination. In *Proceedings of the 12th International Conference on Control, Automation, Robotics and Vision*, Guangzhou, China, pp. 100–105, 2012.

[23] P. Andry, P. Gaussier, J. Nadel, B. Hirsbrunner. Learning invariant sensorimotor behaviors: A developmental approach to imitation mechanisms. *Adaptive Behavior*, vol. 12, no. 2, pp. 117–140, 2004.

[24] A. Shademan, A. M. Farahmand, M. Jagersand. Towards learning robotic reaching and pointing: An uncalibrated visual servoing approach. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, Kelowna, BC, Canada, pp. 229–236, 2009.

[25] F. Nori, L. Natale, G. Sandini, G. Metta. Autonomous learning of 3d reaching in a humanoid robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, pp. 1142–1147, 2007.

[26] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, A, Takanishi. Learning the reachable space of a humanoid robot: A bio-inspired approach. In *Proceedings of IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, Rome, Italy, pp. 1148–1154, 2012.

[27] F. Chao, L. Hu, M. Shi, M. Jiang. Robotic 3D reaching through a development-driven double neural network architecture. In *Proceedings of the* 6th *International Conference on Intelligent Systems and Knowledge Engineering, Knowledge Engineering and Management in Advances in Intelligent and Soft Computing*, Springer, pp. 179–184, 2011.

[28] F. Chao, H. Lin, M. Jiang, L. Zhou. A developmental constraint driven approach to developmental robotic hand-eye coordination. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China, pp. 1848–1853, 2012.

[29] L. Jamone, B. Damas, N. Endo, J. Santos-Victor, A. Takanishi. Incremental development of multiple tool models for robotic reaching through autonomous exploration. *Paladyn*, vol. 3, no. 3, pp. 113–127, 2012.

[30] Q. Meng, M. H. Lee. Error-driven active learning in growing radial basis function networks for early robot learning. *Neurocomputing*, vol. 71, no. 7–9, pp. 1449–1461, 2008.

[31] N. E. Berthier, R. L. Carrico. Visual information and object size in infant reaching. *Infant Behavior and Development*, vol. 33, no. 4, pp. 555–566, 2010.

[32] N. E. Berthier. The syntax of human infant reaching. In *Proceedings of the* 8th *International Conference on Complex Systems, Unifying Themes in Complex Systems Volume VIII*, NECSI Knowledge Press, pp. 1477–1487, 2011.

[33] R. K. Clifton, P. Rochat, D. J. Robin, N. E. Berthier. Multimodal perception in the control of infant reaching. *Journal of Experimental Pschology: Human Perception and Performance*, vol. 20, no. 4, pp. 876–886, 1994.

[34] J. C. Houk. Action selection and refinement in subcortical loops through basal ganglia and cerebellum. *Modelling Natural Action Selection*, Chart 10, A. K. Seth, J. Bryson Ed., Cambridge: Cambridge University Press, 2010.

[35] F. Dandurand, T. R. Shultz. Connectionist models of reinforcement, imitation, and instruction in learning to solve complex problems. *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 110–121, 2009.

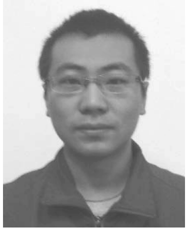
[36] M. H. Lee, Q. Meng. Psychologically inspired sensory-motor development in early robot learning. *International Journal of Advanced Robotics Systems*, vol. 2, no. 4, pp. 325–333, 2005.

[37] Q. Meng, M. H. Lee. Novelty and habituation: The driving forces in early stage learning for developmental robotics. *Neural learning for intelligent robotics, LNCS*, pp. 315–332, 2005.

Fei Chao received his B. Sc. degree in mechanical engineering from Fuzhou University, China in 2004, M. Sc. degree with distinction in computer science from University of Wales, UK in 2005, and Ph. D. degree in robotics from Aberystwyty University, Wales, UK in 2009. He was a research associate under the supervision of Professor Mark H. Lee at Aberystwyth University from 2009 to 2010. He is currently an assistant professor in Cognitive Science Department at Xiamen University, China. He has published about 20 peer-reviewed journal and conference papers. He is a member of IEEE.

His research interests include developmental robotics, machine learning, and optimization algorithms.

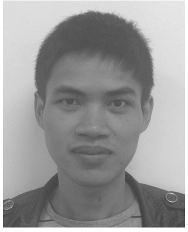
E-mail: fchao@xmu.edu.cn



Xin Zhang received his B. Sc. degree from Xiamen University, China in 2009. He is currently a postgraduate student majoring in artificial intelligence from Cognitive Science Department, Xiamen University, China.

His research interests include artificial neural networks and developmental robotics.

E-mail: shenshikexmu@sohu.com



Hai-Xiong Lin received his B. Sc. degree from Department of Mathematics and Physics, Xiamen University of Technology, China in 2009. He has obtained his M. Eng. degree in artificial intelligence in Cognitive Science Department, Xiamen University, China in 2013.

His research interests include robotics, mathematical modelling, and artificial neural networks.

E-mail: lhxno1729@gmail.com

Chang-Le Zhou received his Ph. D. degree from Peking University, China in 1990. Currently, he is a professor of Cognitive Science Department, Xiamen University, director of Fujian Provincial Key Laboratory of Brain-like Intelligent Systems, and director of Laboratory of ArtMind and Computation. He is also an affiliated professor in linguistics and applied linguistics of Humanity College at Zhejiang University, and an affiliated professor of Philosophy Department at Xiamen University, China.

His research interests lie in the areas of artificial intelligence (AI). His scientific contribution to the AI has more to do with machine consciousness and logic of mental self-reflection. And beyond AI project, he also carries out research on a host of other topics including computational brain modeling, computational modeling of analogy and metaphor and creativity, computational musicology and information processing of data regarding traditional Chinese medicine. His philosophical works lie in ancient oriental thoughts of Chinese, such as ZEN, TAO, YI etc., viewed from science.

E-mail: dozero@xmu.edu.cn

Min Jiang received his bachelor and Ph. D. degrees in computer science from Wuhan University, China in 2001 and 2007, respectively. Subsequently as a post-doc in Department of Mathematics at Xiamen University, China. He studied computational logic, artificial intelligence and its applications on cognitive robot. Currently, he is an associate professor in Cognitive Science Department, Xiamen University. He is a senior member of IEEE and serves as a vice chair of the social media subcommittee of IEEE computational intelligence society.

His research interests include intelligent robotics, computational logics and inference engine.

E-mail: minjiang@xmu.edu.cn (Corresponding author)