

# Visuo-Motor Learning for Face-to-face Pass between Heterogeneous Humanoids

Masaki Ogino <sup>a,\*</sup>, Masaaki Kikuchi <sup>a</sup>, Minoru Asada <sup>a,b</sup>

<sup>a</sup>*Dept. of Adaptive Machine Systems*

<sup>b</sup>*HANDAI Frontier Research Center*

*Graduate School of Engineering, Osaka University,  
2-1 Yamadaoka, Suita, Osaka, Japan*

---

## Abstract

Humanoid behavior generation is one of the most formidable issues due to its complicated structure with many degrees of freedom. This paper proposes a controller for a humanoid to cope with this issue. A given task is decomposed into a sequence of modules first, each of which consists of a set of module primitives that have control parameters to realize the appropriate primitive motions. Then, these parameters are learned by sensori-motor maps between visual information (flow) and motor commands. The controller accomplishes a given task by selecting a module, a module primitive in the selected module, and its appropriate control parameters learned in advance. A face-to-face ball pass in a RoboCup context is chosen as an example task (To the best of our knowledge, this is the first trial.). The corresponding modules are approaching a ball, kicking a ball to the opponent, and trapping a ball coming to the player. In order to show the validity, the method is applied to two different humanoids, independently, and they succeeds in realizing the face-to-face pass more than three rounds.

*Key words:* humanoid, optic flow, face-to-face pass, sensorimotor map

*PACS:*

---

## 1 Introduction

Recent progress of humanoid robots have been attracting many people for their performances of human like behaviors. Although many skills have been

---

\* Corresponding author.

*Email address:* ogino@er.ams.eng.osaka-u.ac.jp (Masaki Ogino ).

*URL:* <http://www.er.ams.eng.osaka-u.ac.jp/> (Masaki Ogino ).

realized in humanoid such as standing up [7], falling down safely [2], and even running [4], the level of the performance is still far from human behaviors on object operation or cooperation with other agents.

In order to generate the skillful performance adaptively, it is inevitable for a humanoid to acquire the sensorimotor mapping, the causal relationship between a self-induced motion and changes in sensory data. However, the mapping between motions and sensors have been often given by the designer [11][13], which causes inflexible response against environmental changes. It is expected that a humanoid obtains the environmental model through the interactions with its environment.

Optical flow has been used to learn the sensorimotor mapping for obstacle avoidance planned by the learned forward model [8] or by finding obstacles that show different flows from the environments using reinforcement learning [10]. Also, it is used for object recognition by active touching [1]. In these studies, the robots have much fewer DoFs than humanoids, therefore it is difficult to directly apply their methods to realize various kinds of humanoid behaviors. One solution for this problem is to decompose a humanoid behavior into basic motion primitives and to acquire the sensorimotor mapping in each motion primitive. This makes it possible for a robot to learn the sensorimotor mapping on line because in each motion primitive the relationship between the motion parameters and the sensor values is usually much simpler than in the case of multiple motions.

This paper presents a method of visuo-motor learning for behavior generation of humanoids, and, as an example task, passing a ball between two different humanoids (face-to-face pass) [6] is realized based on the sensorimotor mappings of motion primitives. The task is decomposed into three basic motion modules: *trapping a ball*, *approaching a ball* and *kicking a ball to the opponent*. Each motion module can be further decomposed into several motion primitives, each of which has parameters to control the motion trajectory. The sensorimotor mapping is learned as the forward and inverse relationships between these parameters and optic flow information in each motion. The acquired sensorimotor maps are used to select the appropriate motion primitive and its parameters to realize the desired pathway or destination in the robot's view given by the designer.

The rest of the paper is organized as follows. Section 2 introduces an overview of our proposed system. Section 3 provides the details of each module for "passing a ball" task. Section 4 shows experimental results of the task that need to use integrated modules. Finally discussions and concluding remarks are given.

## 2 Task, Robot, and Environment

### 2.1 Robot platforms

Fig. 1 shows biped robots used in the experiments, HOAP-1, HOAP-2, and their on-board views. HOAP-1 is 480 [mm] in height and about 6 [kg] in weight [9]. It has a one-link torso, two four-link arms, and two six-link legs. The other, HOAP-2 (a successor of HOAP-1), is 510 [mm] in height and about 7 [kg] in weight. It has two more joints in neck and one more joint at waist that HOAP-1 does not have. Both robots have four force sensing register (FSRs) in each foot to detect reaction force from the floor and a CCD camera with a fish-eye lens (HOAP-1) or semi-fish-eye lens (HOAP-2).

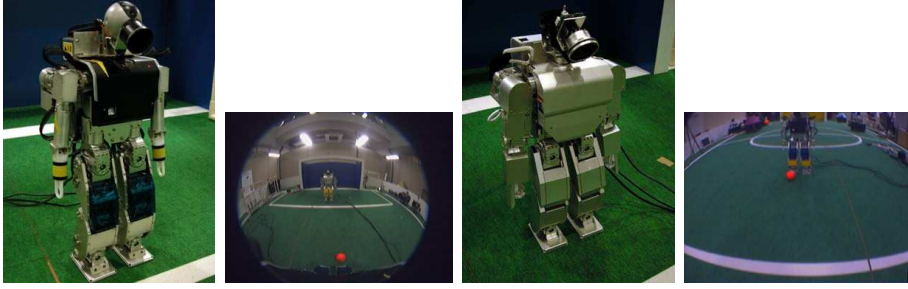


Fig. 1. HOAP-1 with fish-eye lens and HOAP-2 with semi-fish-eye lens

These robots detect objects in the environments by colors. In this experiment, a ball is colored orange, and the knees of the opponent robot are colored yellow. The centers of these colored regions in the images are recorded as the detected positions.

### 2.2 Task and Assumptions

”Face-to-face pass” can be decomposed into following three skills:

- (1) approaching a ball to kick,
- (2) kicking a ball to the opponent, and
- (3) trapping a ball coming to the player

Moreover, each skill consists of more primitive motions; walking to right, straight, left direction and so on. In our proposed controller, each skill is encapsulated as one *motion module* which consists of *motion primitives* for realizing the primitive motions, as shown in Fig. 2. A robot should select an appropriate motion module and an primitive at each moment. There are obvious distinctions between the selection of a motion module and that of a

primitive. In a face-to-face pass task, the sequence of motion modules is deterministic to some extent, and the module to be selected next strongly depends on the previous one. For example, after approaching, kicking module should be selected. Therefore, a simple state machine is used for the module selector, as introduced in section 4.

On the other hand, the primitive to be selected next depends on the situation, rather than the primitive selected previously. Primitives are more independent from each other than the modules. Therefore, the primitive selector selects an appropriate primitive by predicting and evaluating each primitive based on the sensorimotor maps that is learned in advance.

### 2.3 Visuo-motor learning

Different modules need different mapping. For example, to trap a ball appropriately, the robots must estimate when and where a ball comes relative to its own body. To approach a kicking position, the robot should know the causal relationship between the walking parameters and the positional change of the objects in its image. Further, to kick a ball to the opponent, the robot must know the causal relationship between the kicking parameters and the direction of the rolling ball.

However, we can formulate the sensorimotor mapping as the forward and inverse relationships between motion parameters and the sensor values. Let the motion flow vector be  $\Delta \mathbf{r}(t+1)$  at the position  $\mathbf{r}(t)$  in the robot's view when a robot takes a motion,  $a(t)$ . The relationships between them can be written,

$$\Delta \mathbf{r}(t+1) = f(\mathbf{r}(t), a(t)), \quad (1)$$

$$a(t) = g(\mathbf{r}(t), \Delta \mathbf{r}(t+1)), \quad (2)$$

where  $\Delta \mathbf{r}(t+1)$  is difference between the current position vector  $\mathbf{r}(t+1)$  and the previous position vector  $\mathbf{r}(t)$ . The latter function is useful to determine the motion parameters after planning the motion pathway in the image. However, it is difficult to determine a unique motion to realize a certain motion flow because different primitives can produce the same image flow by adjusting motion parameters. Then, we separate the description of the relationship between the parameters in each primitive and the image flow as follows.

$$\mathbf{a}^i = (p_1^i, \dots, p_n^i)^T = g_p^i(\mathbf{r}, \Delta \mathbf{r}) \quad (3)$$

$$\Delta \mathbf{r} = f^i(\mathbf{r}, \mathbf{a}^i), \quad (4)$$

where  $\mathbf{a}^i$  is a parameter vector of the  $i$ -th primitive, and the "p" suffix means that this inverse relationship function associates the *parameters* of the  $i$ -th primitive with image flows. We use neural networks to learn these relationships.

The details of the primitive selector and sensorimotor maps in each motion module are given in the next section.

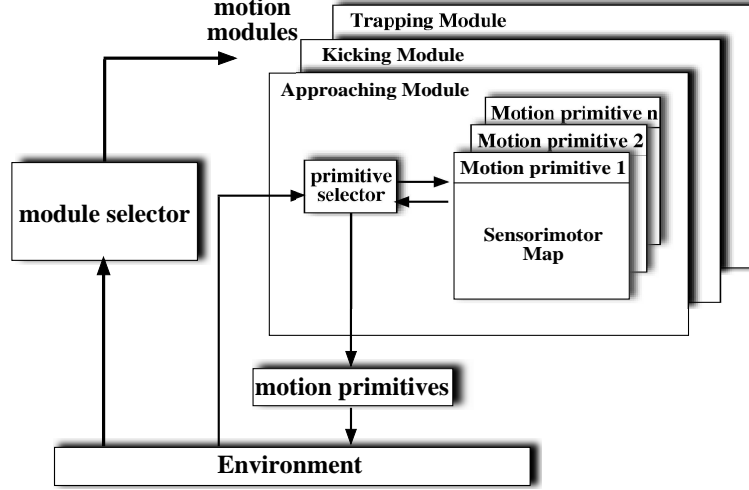


Fig. 2. A system overview

### 3 Motor Learning Based on Optic Flow Information

#### 3.1 Approaching the ball

Approaching a ball is the most difficult task among the three modules because this task involves several motion primitives each of which has parameters to be determined. These motions yield characteristic patterns of image flows depending on the values of the parameters which change continuously. The *Approaching* module learns the sensori-motor maps, and makes use of them for determining an appropriate motion primitive and those parameter values (Fig. 3).

The relationship between motions and image flows is described in two levels: the relationship between the motion primitives and the image flows, and the relationship between the parameters in each primitive and the image flows, as follows:

$$\mathbf{m} = g_m(\mathbf{r}, \Delta \mathbf{r}), \quad (5)$$

$$\mathbf{a}^i = (p_1^i, p_2^i)^T = g_p^i(\mathbf{r}, \Delta \mathbf{r}), \quad (6)$$

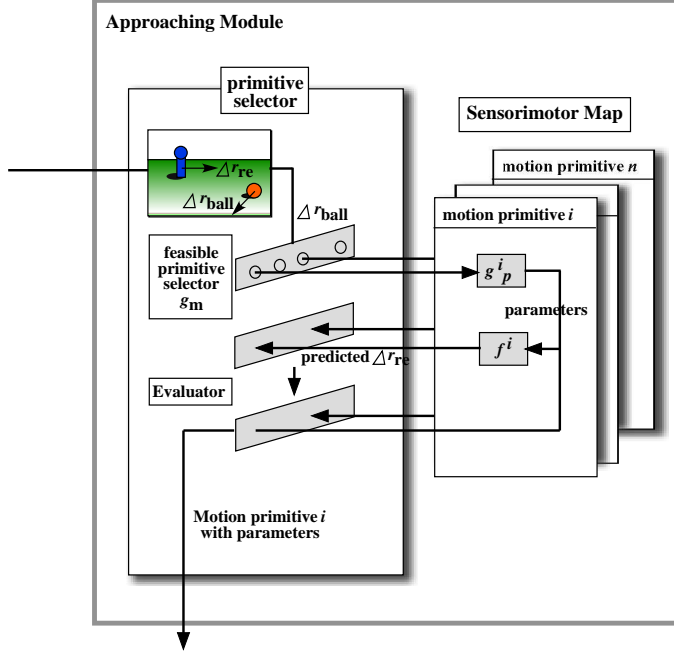


Fig. 3. An overview of the approaching module

$$\Delta \mathbf{r} = f^i(\mathbf{r}, \mathbf{a}^i), \quad (7)$$

where the suffix "m" means that the inverse relationship function associates the *motion primitive* with image flow.  $\mathbf{m}$  is a motion primitive vector whose  $i$ -th element indicates the effectiveness of  $i$ -th primitive to generate the given flow, and  $\mathbf{a}^i = (p_1^i, p_2^i)^T$  is the parameter vector of the  $i$ -th motion primitive. The primitives related to *Ball Approaching* module consists of six primitives; *forward walk* (left and right), *curve walk* (left and right), and *side step* (left and right). Each primitive has two parameters which have continuous values, as shown in Fig. 4. We made these walking primitives based on rhythmic walking algorithm [12] [16].

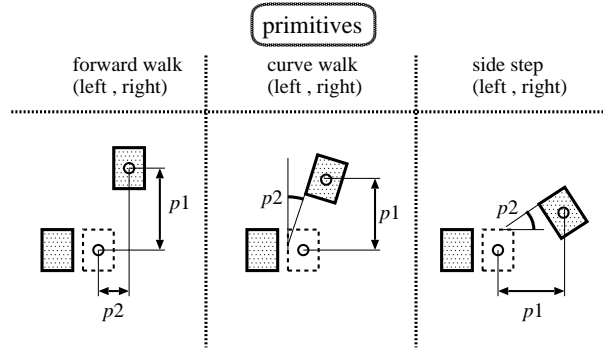


Fig. 4. Motion primitives and parameters for approaching

Images are recorded every step and the image flow is calculated by block matching between the current image and the previous one. The windows for calculating flows are 24 blocks in one image as shown in Fig. 5.



Fig. 5. An example of an optic flow in the robot's view

The approaching module determines the appropriate parameter values through the following 3 steps.

**1st step: Planning the desired optic flow** First, the system sets the desired optic flow to be realized so that the robot can approach the appropriate kicking position.

**2nd step: Selecting motion primitives by the inverse function** Given the desired motion pathway in the robot's view, we can select appropriate primitive by  $g_m$ , and determine the parameters of the selected primitive by  $g_p^i$  based on the learned relationships among the primitives, their parameters, and flows.

**3rd step: Evaluating motion primitives by the forward function** If the desired image flow yields several motion primitives, the preferred primitive is determined by an evaluation function.

The forward function,  $f^i$ , and inverse functions,  $g_p^i$ , which correlates the relationship between the parameters in each primitive and the image flow, are realized by a simple neural network. The neural network in each primitive is trained by the backpropagation algorithm so that it outputs the parameters when the flow vector and the positional vector in the image are input.

### *Planning the desired optic flow*

In the first step, the approaching module determines the desired optic flows to be realized for the ball and the receiver: the flow vector for the ball,  $\mathbf{s}_{ball}$ , is set as the vector from the current position of a ball to the desired position (kicking position) in the robot's view, and the flow vector for the receiver,  $\mathbf{s}_{re}$ , is set as the horizontal vector from the current position to the vertical center line. The next desired optic flow of the ball to be realized,  $\tilde{\mathbf{s}}_{ball}$ , is calculated based on these desired optic flows,

$$n_{step} = \|\mathbf{s}_{ball}\| / \Delta r_{max}, \quad (8)$$

$$\tilde{\mathbf{s}}_{ball} = \mathbf{s}_{ball} / n_{step}, \quad (9)$$

where  $\Delta r_{max}$  is the maximum length of the experienced optic flow. This reference vector is input to the primitive selector,  $g_m$ , and the candidate primitives which can output the reference vector are activated.

#### *Selecting motion primitives by the inverse function, $g_m$*

The inverse function to select appropriate primitives which realize the desired optic flow is implemented with the self organizing map (SOM) [14]. In learning phase, all of the data sets of the flow and its positional vector in the image while walking using various motion primitives with various parameter values,  $(\mathbf{r}, \Delta \mathbf{r})$ , are classified by SOM, which consists of 225 ( $15 \times 15$ ) representational vectors. After organization, the indices of motion primitives are attributed to each node on SOM with a motion primitive vector,  $\mathbf{m}_{i,j}$ , in which the  $k$ -th element is assigned 1 if the  $k$ -th motion primitive can generate the optical flow that is recorded as the representational vector at  $(i, j)$  node on SOM. For example, if the 2nd and 3rd motion primitives can generate the optic flow that is recorded as a representational vector at  $(i, j)$  node on SOM, the motion primitive vector at  $(i, j)$  can be described as

$$\mathbf{m}_{i,j} = (0, 1, 1, 0, 0, 0). \quad (10)$$

Fig. 6 shows the classified image vector (the figure at the left side) and the distribution of motion primitives that can generate the the representational vector as optic flow at each node.

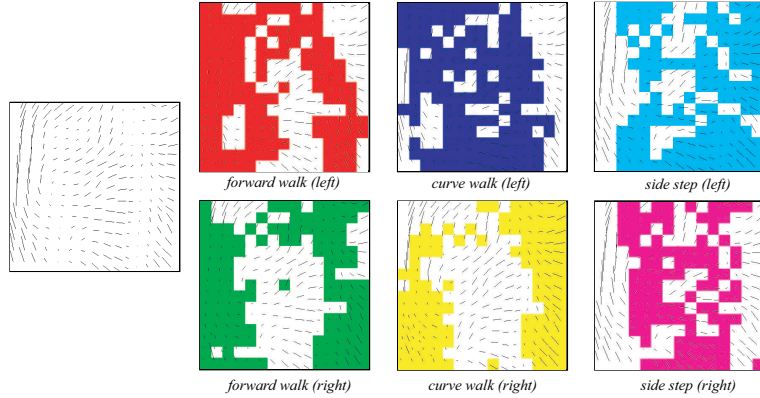


Fig. 6. Distribution of motion primitives on the SOM of optic flows

Once the desired optic flows are determined by planning process, the node that has the closest representational vector to the desired one is selected. Referring to the motion primitive vector of the selected node on SOM, the

appropriate primitive can be selected. The parameters of the selected primitive are determined by the function  $g_p^i$ ,

$$\mathbf{a}^i = g_p^i(\mathbf{r}_{ball}, \tilde{\mathbf{s}}_{ball}), \quad (11)$$

where  $\mathbf{r}_{ball}$  is the current ball position in the robot's view.

#### *Evaluating motion primitives by the forward function*

When the primitive selector outputs several candidates of primitives, the evaluation function depending on the task,  $V(m_i)$ , determines the preferred primitive, because robots have to not only approach a ball but also take an appropriate position to kick a ball to the other. For that, we set the evaluation function as follows,

$$\begin{aligned} V(m_i) = & \|\tilde{\mathbf{s}}_{ball} - f^i(\mathbf{r}_{ball}, \mathbf{a}^i)\| \\ & + k\|\mathbf{s}_{re} - n_{step}f^i(\mathbf{r}_{re}, \mathbf{a}^i)\|, \\ P = & \arg \min_{i \in \text{primitives}} V(m_i) \end{aligned} \quad (12)$$

where  $k$  is a constant,  $\mathbf{r}_{re}$  is the current position of the receiver in the robot's view, and  $P$  is the selected primitive.

#### *Experimental results*

Fig. 7 shows experimental results of approaching a ball. A robot successfully approach a ball so that the hypothetical opponent (a poll) comes in front of it.

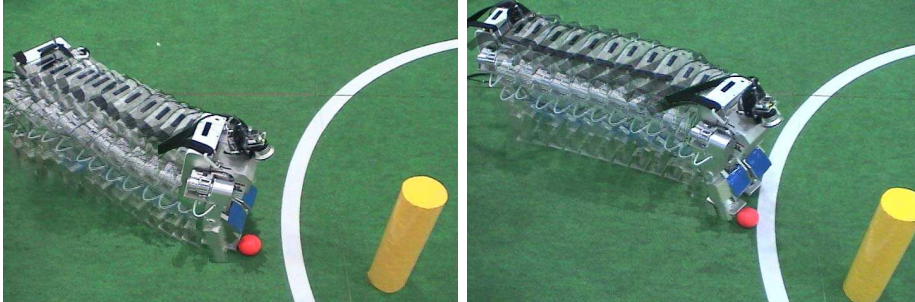


Fig. 7. Experimental results of approaching a ball

### 3.2 Ball Kicking to the Opponent

It is necessary for the robots to kick a ball to the receiver very precisely because they cannot sidestep quickly. We correlate the parameter of kicking motion with the trace of the kicked ball in the robot's view so that they can kick to each other precisely. Fig. 8 shows a proposed controller for kicking.

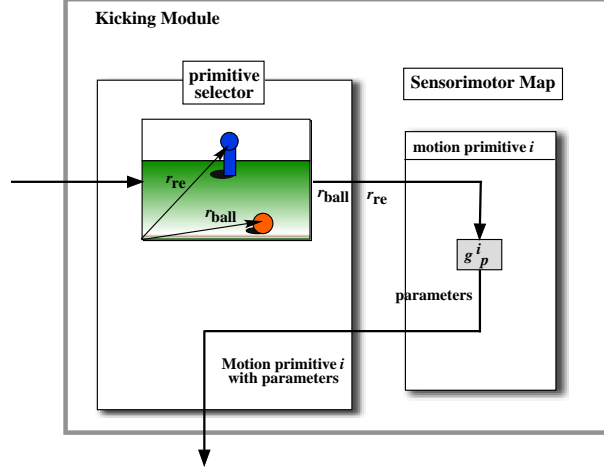


Fig. 8. An overview of the kicking module

The kicking parameter is the hip joint angle shown in Fig. 9 (a). A quick motion like kicking changes its dynamics depending on its parameter value. The sensor feedback from the floor reaction force sensors is used to stabilize the kicking motion. The displacement of the position of the center of pressure (CoP) in the support leg is used as feedback to the angle of the ankle joint of the support leg (see Fig. 9(b)), as follows,

$$\Delta\theta_4 = -K(y_{CoP} - y_{CoP}^0), \quad (13)$$

where  $\theta_4$  is the ankle joint of the support leg,  $K$  is the feedback gain.  $y_{CoP}$  and  $y_{CoP}^0$  are the current position and the desired position of CoP, respectively. Fig. 9 (c) shows the effectiveness of the stabilization of the kicking motion.

The ball trace from the robot's view is sensitive to both the initial ball position and the kicking parameter. To learn the relationship of these variables, a neural network is trained in the environment where the poll (10 [cm]) is put about 1 [m] in front of the robot (Fig. 9 (a)). The trace of the ball (the effects of the self motion is subtracted) is recorded every 100 [msec], and the weights in the neural network are updated by the backpropagation algorithm every one trial. Fig. 9 (b) shows the time course of error distance between target poll position and kicked ball in the robot's view. It shows that the error is reduced rapidly within 20 [pixel], which is the same size of the width of the target poll. Fig. 11 shows the kicking performance of the robot.

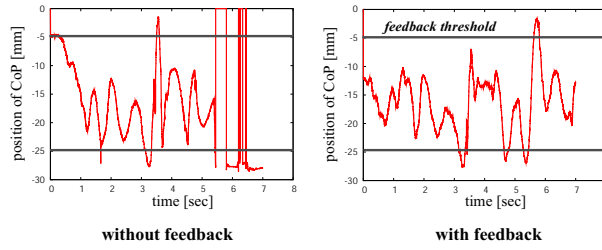
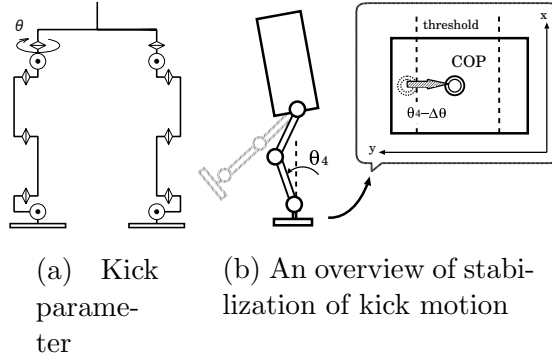


Fig. 9. The parameter and the stabilization of kicking

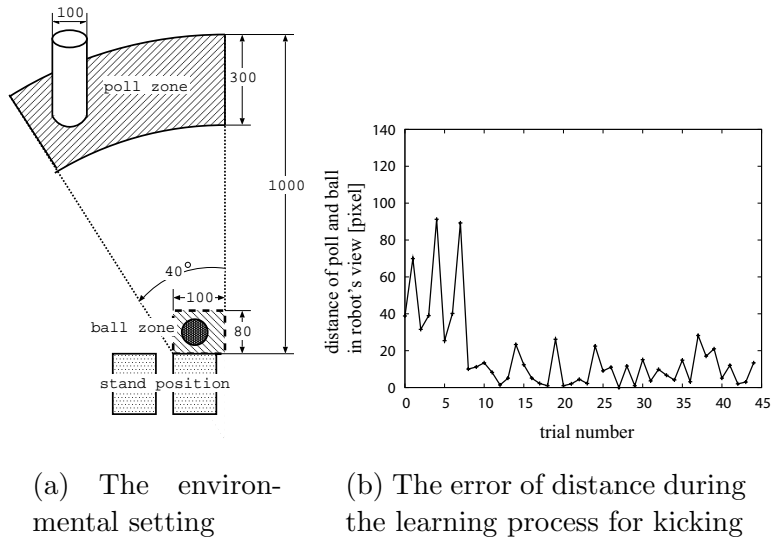


Fig. 10. The environmental setting and the learning curve for kicking

### 3.3 Ball Trapping

Fig. 12 shows an overview of trapping module. Robots learn the relationship between the position of the foot in robot's view and the trap parameter which



Fig. 11. An experimental result of kicking a ball to the poll affects the position of the foot, to acquire the skill to trap a coming ball.

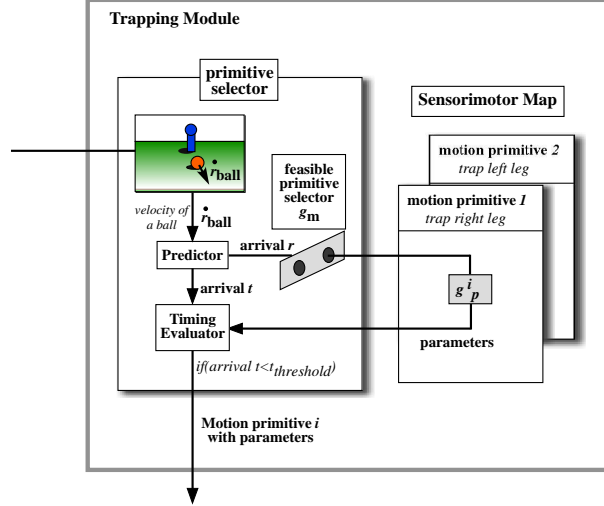


Fig. 12. An overview of the trapping module

Fig. 14 shows the trapping motion by HOAP-2 acquired by the method described below. In order to realized such a motion, the robot has to predict the position and the arrival time of a ball from its optic flow captured in the robot view. For that purpose, we use a neural network which learns the causal relationship between the position and an optic flow of the ball in the observed image of a robot and the arrival position and time of the coming ball. This neural network is trained by the data in which a ball is thrown to a robot from the various positions. Fig. 13 shows several prediction results of the neural network after learning.  $\Delta x$  [pixel] and  $\Delta t$  [sec] indicates the errors of the arrival position and the time predicted at each point (every 0.3[sec]) in the robot's view.  $T$  denotes a duration of the ball rolling. Based on this neural network, the robots can activate the trapping motion primitive with the appropriate leg (right or left) at the appropriate timing (Fig. 14).

#### 4 Integration of the Modules for Face-to-face Pass

To realize passing a ball between two heterogeneous humanoids, the basic modules described in the previous sections are integrated by a simple rule as shown in Fig. 15. This is possible because, in a face-to-face pass task, the sequence of motion modules is deterministic to some extent, and the next

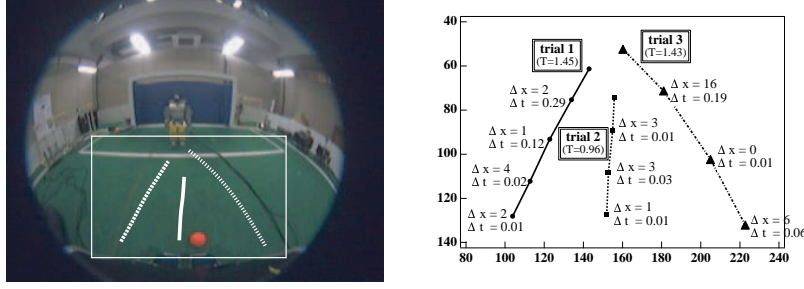


Fig. 13. The prediction of the position and time of a coming ball

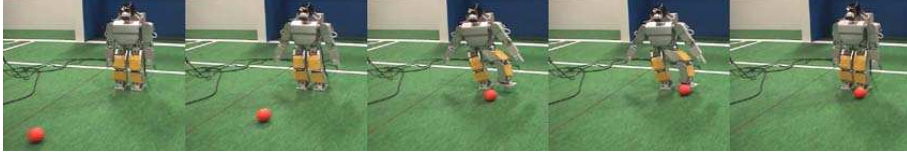


Fig. 14. An experimental result of a trapping module

module to be selected strongly depends on the previous module. For example, after approaching, kicking module should be selected.

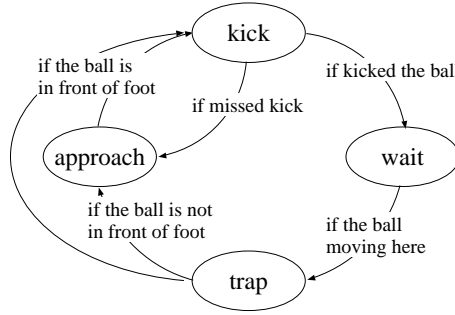
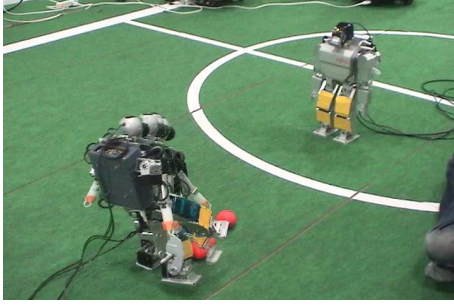


Fig. 15. The rule for integrating motion modules

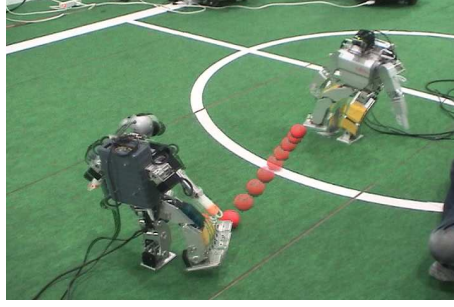
Fig. 16 shows the experimental result. Two humanoids with different body and different camera lens realize the appropriate motions for passing a ball to each other based on their own sensorimotor mapping. The passing lasts more than three round trips.

## 5 Discussion

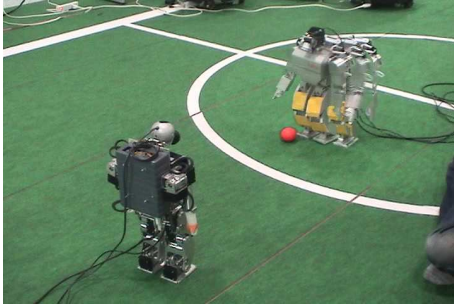
A humanoid has many degrees of freedom, which makes it more difficult to make motions than in wheel robots. In this paper, we encapsulated humanoid behaviors into motion modules, each of which consists of multiple motion primitives. This method can be related to the biological neural circuits for locomotion, CPG (Central Pattern Generator) [3]. In cats, neurons for various types of locomotions are found in the midbrain and it is believed that the deviations of one locomotion is performed by the input from the higher central



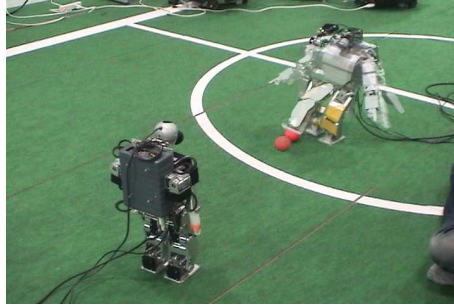
(a) kicking



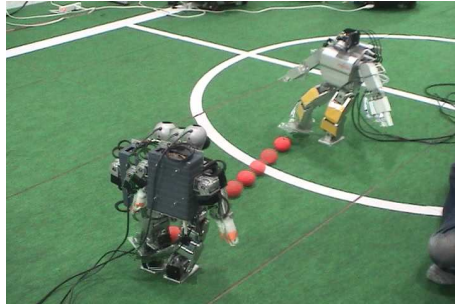
(b) trapping



(c) approaching



(d) kicking



(e) trapping

Fig. 16. An experimental result of passes between two humanoids

nervous system. Thus, nature seems to solve the problem of DoFs to some extent in the same way as the proposed model in this paper. The biological model for locomotion is proposed by Taga [15]. They demonstrate that the simple input to the CPG circuits makes it possible for a humanoid model to walk over obstacles in the computer simulation environment.

The proposed method makes fully use of the relationship between the motions and the optical flows of the environment in two aspects. First, using the model, the robot can estimate the resultant position of the ball and the opponent after

the motion. If the set of the positions of a ball and a goal are taken as one state, then it becomes very difficult to make a state transition model because the robot must examine all the possible combinations of configuration of the ball and the opponent. In this study, instead of adopting such a state, the simple optical flows of the environment during the motion are used, which are irrelevant to the configuration of objects around the robot. And this sensory motor model is applied to each object, which enables the system to make a model as the combinations of the existing model. This accelerates the learning.

The second advantage is compensation of self motion. Using the self-centered coordinates involves the problem of the effects of the self motion. The kicked ball trace should be compensated by the self kicking motion. The ball position to be approached should be monitored by the compensated self walking motion. The trapping foot position should be recognized with compensation of the self trapping motion. In our experiments, those data are compensated with the learned optic flows of the static environment of each motion, although it is not mentioned explicitly in this paper.

In both aspects, it is the designer that determines to use the optical flows that are learned in the static environment for compensating the position of objects. It is not always the case that a model learned in one situation can be applied in another situation. For example, the environmental flow cannot be applied for prediction of a moving ball. To solve this problem, it is necessary to build a motion model automatically and apply an appropriate model to the current situation. Such a system might be possible by adapting a model like MOSAIC to making object models [5] in which an appropriate module is selected by the accuracy of forward prediction of each module.

## 6 Conclusions

In this paper, the robots learn the sensorimotor mapping between optic flow information and their own parameters of motion primitives. Acquiring basic modules for passing a ball is achieved using the sensorimotor mapping. In each module, optic flow information is correlated with the parameters. Through this correlation, a humanoid robot can obtain the sensorimotor mapping to realize the desired modules. The experimental results show that a simple neural network quickly learns and models well the relationship between optic flow information and parameters of each motion primitive.

## Acknowledgement

This research was supported by the “Robot Brain Project” under the Core Research for Evolutional Science and Technology (CREST program) of the Japan Science and Technology Corporation.

## References

- [1] P. Fitzpatrick, First contact: an active vision approach to segmentation, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 2161–2166.
- [2] K.Fujiwara, F.Kanehiro, H.Saito, S.Kajita, K.Harada, and H. Hirukawa, Falling motion control of a humanoid robot trained by virtual supplementary tests, in: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, 2004, pp. 1077-1082.
- [3] S. Grillner, Neurobiological bases of rhythmic motor acts in vertebrates, *Science* 228 (1985) 143-149.
- [4] S. Kajita, T. Nagasaki, K.Kaneko, K.Yokoi, and K.Tanie, A hop towards running humanoid biped, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2004, pp. 629-635.
- [5] D.M.Wolpert and M.Kawato, Multiple paired forward and inverse models for motor control, *Neural Networks* 11 (1998) 1317-1329.
- [6] H. Kitano and M. Asada, The RoboCup humanoid challenge as the millennium challenge for advanced robotics, *Advanced Robotics* 13 No.8 (2000) 723-736.
- [7] Y. Kuniyoshi, Y. Ohmura, K. Terada, and A. Nagakubo, Dynamic roll-and-rise motion by an adult-size humanoid robot, *International Journal of Humanoid Robotics* 1 No.3 (2004) 497-516.
- [8] K. F. MacDorman, K. Tatani, Y. Miyazaki, M. Koeda, and Y. Nakamura, Protosymbol emergence based on embodiment: Robot experiments, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2001, pp. 1968–1974.
- [9] Y. Murase, Y. Yasukawa, K. Sakai, etc. Design of a compact humanoid robot as a platform. in: Proceedings of the 19th Conference of Robotics Society of Japan, 2001, pp. 789-790.
- [10] T. Nakamura and M. Asada, Motion sketch: acquisition of visual motion guided behaviors, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1995, pp. 126–132.

- [11] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba, and H. Inoue, Walking control system of a humanoid for tracking a moving object with estimate of the target motion, in: Proceedings of the 3rd IEEE International Conference on Humanoid Robots, 2003.
- [12] M. Ogino, Y. Katoh, M. Aono, M. Asada, and K. Hosoda, Reinforcement learning of humanoid rhythmic walking parameters based on visual information, *Advanced Robotics* 18 No. 7 (2004) 677–697.
- [13] J. F. Seara, K. Strobl, and G. Schmidt, Information management for gaze control in vision guided biped walking, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 31–36.
- [14] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, 1989.
- [15] G. Taga, A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance, *Biological Cybernetics* 78 (1998) 9–17.
- [16] K. Tsuchiya, K. Tsujita, K. Manabu, and S. Aoi, An emergent control of gait patterns of legged locomotion robots, in: Proceedings of the Symposium on Intelligent Autonomous Vehicles, 2001, pp. 271-276.