Proceedings of the 2008 IEEE
International Conference on Robotics and Biomimetics
Bangkok, Thailand, February 21 - 26, 2009

Active Motor Babbling for Sensorimotor Learning*

Ryo Saegusa, Giorgio Metta, and Giulio Sandini
Robotics, Brain and Cognitive Sciences Department
Italian Institute of Technology
Via Morego 30, 16163 Genoa, Italy
ryos@ieee.org, pasa@liralab.it, giulio.sandini@iit.it

Sophie Sakka
Laboratory of Solids Mechanics
University of Poitiers
BP30179-86962 Futuroscope Chasseneuil Cedex, France
sophie.sakka@lms.univ-poitiers.fr

***Abstract*—For a complex autonomous robotic system such as a humanoid robot, motor-babbling-based sensorimotor learning is considered an effective method to develop an internal model of the self-body and the environment autonomously. In this paper, we propose a method of sensorimotor learning and evaluate it performance in active learning. The proposed model is characterized by a function we call the " confidence", and is a measure of the reliability of state prediction and control. The confidence for the state can be a good measure to bias the next exploration strategy of data sampling, and to direct its attention to areas in the state domain less reliably predicted and controlled. We consider the confidence function to be a first step toward an active behavior design for autonomous environment adaptation. The approach was experimentally validated using the humanoid robot James.**

***Index Terms*—sensorimotor learning, neural networks, state prediction, humanoid robot, confidence**

I. Introduction

Learning in robotics is one practical solution allowing an autonomous robot to perceive its body and the environment. As discussed in the context of the *frame problem* [1], the robot's body and the environment are generally too complex to be modeled. Even if the kinematics and the dynamics of the body are known, a real sensory input to the body often differs from one derived from a theoretical model, because sensor input is always influenced by interaction with the environment. For instance, when we grasp an object, the physical parameters of our arm, such as its mass and momentum, differs from the nominal state depending on the grasped object. Moreover, it is difficult to evaluate all potential variations in advance, since real data can vary quite a lot and the behavior of the external environment is not necessarily controlled by the robot. On the other hand, learning approaches provide a data-driven solution: the robot explores the environment and extracts knowledge to build an internal model of the body and the environment.

Learning-based motor control systems are well studied in the literature [2][3][4][5][6][7]. Haruno et al. proposed a modular control approach [3], which couples a forward model (state predictor) and an inverse model (controller). The forward model predicts the next state from the current state and a motor command (an efference copy), while the inverse model generates a motor command from the current state and the predicted state. Even if a proper motor command is unknown, the feedback error learning procedure (FEL) provides a suitable approximation [4]. The prediction error contributes to gate learning of the forward and inverse models, and to weight output of the inverse models for the final motor command. Motor prediction, based on a copy of the motor command, compensates for delays and noise in the sensorimotor system. Moreover, motor prediction allows differentiating self-generated movements from externally imposed forces/disturbances [5][6].

Learning-based perception is applicable not only for motor control but also to model the environment using multiple sensorial modalities, such as vision, audition, touch, force/torque, and acceleration sensing. In our earlier approach, we developed a learning system aimed at predicting future sensor data based on current sensor data and motor commands [8]. In the study we explored the possibilities for the robot to detect changes in its body or the environment in an autonomous manner: no other information, such as a kinematic model, was given to the system. Following this concept, we investigated a function called *confidence*, focused on sensory prediction learning [9]. The function of confidence is to quantify inequalities between the predicted state and the real state of the body and the environment.

One of the significant problems in learning is that learning domain is too large to be completely covered, as mentioned at the beginning of this article with frame problem. An efficient learning strategy is necessary to enhance learning speed while keeping its quality high. A random sampling strategy is often considered to be the most robust approach for unknown learning domain. However, the effect of learning is biased by various factors and constraints in the learning domain. In the literature of sensory-motor learning, conventional motor babbling generates random motor commands in joint space to collect learning data in task space, while the data sampling is biased by nonlinear mapping from joint space to task space.

We propose an improvement of the learning strategy: *active motor babbling* based on confidence for the state.

*The work presented in this paper was partially supported by the ROBOTCUB project (IST-2004-004370) and the CONTACT project (NEST-5010), funded by the European Commission through the Unit E5 Cognitive Systems.

notation	variable
$\boldsymbol{s}$	measured sensory input
$\hat{\boldsymbol{s}}$	estimated sensory input
$\hat{\boldsymbol{\delta s}}$	estimated sensory input change
$\boldsymbol{s}^*$	desired sensory input
$\boldsymbol{\delta s}^*$	desired sensory input change
$\boldsymbol{u}$	actuated motor command
$\hat{\boldsymbol{u}}$	estimated motor command
$\boldsymbol{u}^*$	desired motor command
$\Phi(\cdot)$	state prediction function
$\Psi(\cdot)$	state control function
$\chi(\cdot)$	state design function
$\hat{\Phi}(\cdot)$	approximated state prediction function
$\hat{\Psi}(\cdot)$	approximated state control function

TABLE I
VARIABLE AND FUNCTION NOTATION.

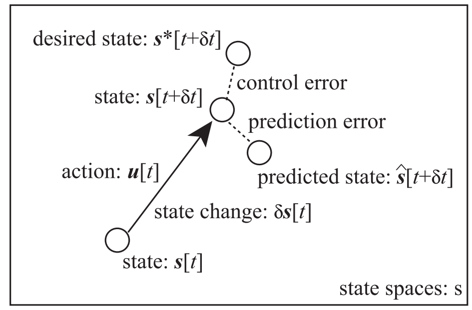


Fig. 1. State transition diagram of the proposed sensorimotor system.

This approach is an extension of [9] to deal with both the problems of state prediction and state control. That is, the current learning evaluation on state prediction and control is applied to the next exploration strategy for data sampling by using acquired state control skills. The exploration strategy focuses data sampling on insufficient parts of learning.

This paper is organized as follows: Section II describes the proposed framework of sensory-motor learning including an introduction of the *confidence* function. Section III describes the experimental results obtained using the humanoid robotic platform James [10]. Finally, Section IV gives the conclusion and outlines some future tasks.

II. METHOD

A. Sensorimotor learning

Fig. 1 illustrates the internal state space of a sensorimotor system. variable notation used in this figure is defined in Table I. Let $\boldsymbol{s}[t] \in R^{N_s}$ denote the sensory input vector from the N_s sensors, and $\boldsymbol{u}[t] \in R^{N_m}$ be the motor command vector for the N_m motors at time t. Here, we assume the sensory input vector as the state vector, and discuss the state space formed by the set of all state vectors. The state is changed by the motor command actuation. Let us assume that the dynamics of $\boldsymbol{s}[t]$ and $\boldsymbol{u}[t]$ can be defined as:

$$\boldsymbol{s}[t+\delta t] = \boldsymbol{s}[t] + \boldsymbol{\delta s}[t], \quad (1)$$
$$\boldsymbol{\delta s}[t] = \Phi(\boldsymbol{s}[t], \boldsymbol{u}[t]), \quad (2)$$
$$\boldsymbol{u}[t] = \Psi(\boldsymbol{s}[t], \boldsymbol{\delta s}[t]). \quad (3)$$

Here, for simplicity, we assume that δt is sufficiently small that a motor command to change the state from $\boldsymbol{s}[t]$ to $\boldsymbol{s}[t+\delta t]$ is unique. In this manner we can model state transitions without considering the problem of kinematic redundancy in the local domain.

The goal of learning is to approximate $\Phi(\cdot)$ and $\Psi(\cdot)$ using data samples acquired through exploration. Let $\hat{\boldsymbol{\delta s}}[t]$ and $\hat{\boldsymbol{u}}[t]$ denote estimated vectors of the sensory input change $\boldsymbol{\delta s}[t]$ and that of the actuated motor command $\boldsymbol{u}[t]$, respectively. $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$ denote the approximations of $\Phi(\cdot)$ and $\Psi(\cdot)$, defined as:

$$\hat{\boldsymbol{\delta s}}[t] := \hat{\Phi}(\boldsymbol{s}[t], \boldsymbol{u}^*[t]), \quad (4)$$
$$\hat{\boldsymbol{u}}[t] := \hat{\Psi}(\boldsymbol{s}[t], \hat{\boldsymbol{\delta s}}[t]), \quad (5)$$

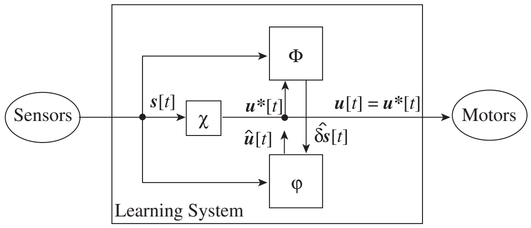
where the estimated state change $\hat{\boldsymbol{\delta s}}[t]$ is used as an input for the estimation of state control. $\boldsymbol{u}^*[t_k]$ is applied to robot as $\boldsymbol{u}[t_k] = \boldsymbol{u}^*[t_k]$. The functions $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$ represent internal sensorimotor dynamics, which can be exploited for state prediction and state control, as shown in the Fig. 2.

In order to collect learning data for these function approximations, the robot must move its body. At the beginning of the learning process, however, the robot does not know how to control its joint movement. Motor babbling gives us a simple solution to this problem: the learning system randomly generates a motor command $\boldsymbol{u}^*[t_k]$, which is an output of the state design function illustrated as $\chi(\cdot)$ in the Fig. 2a. The robot then actuates this motor command as $\boldsymbol{u}[t_k] = \boldsymbol{u}^*[t_k]$, leading to random joint movement. During motor babbling, the learning system stores measured data: $\{\boldsymbol{s}[t_k], \boldsymbol{u}[t_k], \boldsymbol{\delta s}[t_k]\}_{k=1,\cdots,K}$ at each time step: t_k. Let us refer to the above process as the *U-space* motor command generation (Fig.2a). In learning of the functions $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$, $\boldsymbol{s}[t_k]$, $\boldsymbol{u}[t_k]$, and $\boldsymbol{\delta s}[t_k]$ can be used as input vectors of $\boldsymbol{s}[t]$, $\boldsymbol{u}[t]$, and $\boldsymbol{\delta s}[t]$, respectively, while $\boldsymbol{\delta s}[t_k]$ and $\boldsymbol{u}[t_k]$ can be used as target vectors of $\hat{\boldsymbol{\delta s}}[t]$ and $\hat{\boldsymbol{u}}[t]$, respectively.

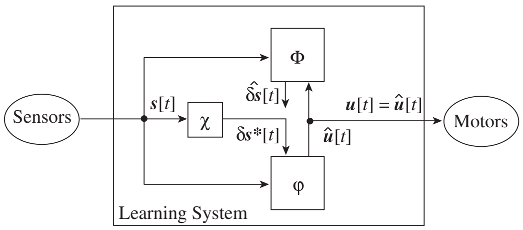
If the learning process is complete, the robot will be able to generate a motor command to reach a desired next state $\boldsymbol{s}^*[t]$, defined as:

$$\hat{\boldsymbol{\delta s}}[t] = \hat{\Phi}(\boldsymbol{s}[t], \hat{\boldsymbol{u}}[t]), \quad (6)$$
$$\hat{\boldsymbol{u}}[t] = \hat{\Psi}(\boldsymbol{s}[t], \boldsymbol{\delta s}^*[t]), \quad (7)$$

where the estimated motor command: $\hat{\boldsymbol{u}}[t]$ is used for actuation of the robot joints as $\boldsymbol{u}[t] = \hat{\boldsymbol{u}}[t]$. This is represented in Fig. 2b. The relations between the current and next state are $\hat{\mathrm{s}}[t+\delta t] = \boldsymbol{s}[t] + \hat{\boldsymbol{\delta s}}[t]$ and $\boldsymbol{s}^*[t+\delta t] = \boldsymbol{s}[t] + \boldsymbol{\delta s}^*[t]$. By using the approximated functions, the robot is able to generate a motor command to collect *interesting* learning samples in state space. Let us refer to the above process as the *S-space* motor command generation (Fig. 2b).



(a) *U-space* motor generation



(b) *S-space* motor generation

Fig. 2. A comparison between passive motor command generation (a) and active motor command generation (b).

B. Confidence for a state

Learning results can be evaluated in terms of the *confidence* for a state. The confidence is based on the state prediction error e_s and motor control error e_u defined as

$$e_s[t] = |\hat{\boldsymbol{\delta s}}[t] - \boldsymbol{\delta s}[t]|, \tag{8}$$
$$e_u[t] = |\hat{\boldsymbol{u}}[t] - \boldsymbol{u}[t]|, \tag{9}$$

when $\boldsymbol{u}[t]$ $(= \boldsymbol{u}^*[t])$ is given by *U-space* motor command generation. If the motor command is given by *S-space* motor command generation, Eqn.(9) cannot be used, since the equation $\hat{\boldsymbol{u}}[t] = \boldsymbol{u}[t]$ is always true, leading to permanent zero control error. In this case, we use the following error vector instead of $e_u[t]$,

$$e_p[t] = |\boldsymbol{\delta s}^*[t] - \boldsymbol{\delta s}[t]|, \tag{10}$$

which gives the error between the measured state and desired state by motor control, meaning the performance error of the state control in the *S-space*.

Let us introduce the Gaussian filtering of $e_s, e_u \in (0, +\infty)$ as a finite scalar variable $c[t] \in [0, 1]$ such as

$$c[t] = \exp\left(-\frac{e_s[t]^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{e_u[t]^2}{2\sigma^2}\right), \tag{11}$$

where the constant σ^2 determines filtering sensitivity (Fig.3). Accumulation of $c[t]$ depends on the state $\boldsymbol{s}[t]$, and provides robust memory of confidence for the state $\boldsymbol{s}[t]$ on prediction and control. Let $C_{\boldsymbol{s}} \in [0, 1]$ denote the *confidence*, working as a temporal moving average of normalized learning error $c[t]$. The update rule of the confidence for $\boldsymbol{s}$ at time t is defined as:

$$C_{\boldsymbol{s}}[t] := (1 - \alpha)C_{\boldsymbol{s}}[t - \delta t] + \alpha c[t], \tag{12}$$

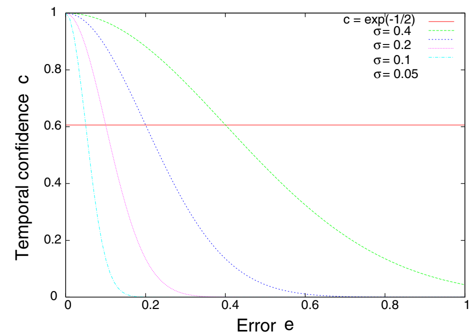


Fig. 3. Temporal confidence vs error. The horizontal coordinates of the intersections between the line $c = \exp(-1/2)$ and Gaussian curves for several values of σ, which adjusts filtering sensitivity.

where the constant parameter: $\alpha \in [0, 1]$ denotes an update weight. $C_{\boldsymbol{s}}[0]$ is initialized as zero at the beginning of the learning process. A high value of $C_{\boldsymbol{s}}$ indicates that knowledge of state dynamics at the state $\boldsymbol{s}$ is reliable.

C. Learning strategy

The sensorimotor learning procedure is divided into two stages: *exploration* and *learning*, as illustrated in Fig.**??**. In the *exploration* stage, the robot generates joints movements (motor babbling) in order to collect learning samples, and evaluates mapping functions optimized in previous *learning* stages. In the *learning* stage, the robot optimizes the mapping functions off-line with the collected learning samples in the previous *exploration* stages. Motor behavior of the robot in the *exploration* stage is generated in *U-space* or *S-space*. If the confidence for the current state at time t is lower than a constant threshold β, described as $C_{\boldsymbol{s}}[t] < \beta$, a motor command is randomly generated in *U-space*. Otherwise, the minimum-confidence state among current eight-neighbor states is given to the robot as the next desired state, and the motor command is generated by the function $\Psi(\cdot)$.

The principal idea of this framework is to exploit confidence derived from past learning experience, and then focus subsequent exploration to collect new learning data of *interest*. Here, we are focusing on increasing learning efficiency. However, we could give other learning strategies to exploit the advantages of active exploration and learning. For instance, if the confidence is high enough, the robot can direct its action by using acquired motor skills toward the state which attracts its attention.

D. Implementation by neural networks

The functions $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$ were approximated using Multi Layer Perceptron (MLP) as shown in Fig.5 [8][11]. MLPs are universal function approximators, whose parameters can be optimized by learning. In our system, we implemented the learning strategy using a three layer MLP and conventional gradient descent. [11].

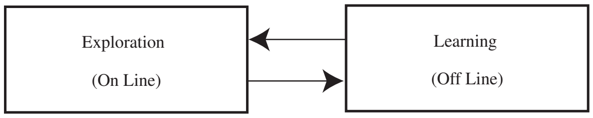


Fig. 4. The proposed learning strategy. A robot explores the environment to collect learning data, and evaluates sensorimotor functions on-line. After exploration, the robot optimizes sensorimotor functions with the collected learning samples off-line. These two processes are repeated alternatingly until the desired performance is reached.

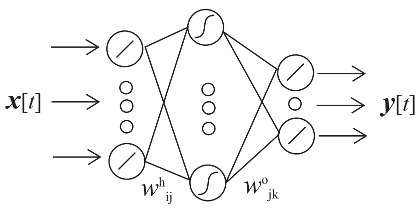


Fig. 5. The Multi Layer Perceptron (MLP) used for approximation of the state prediction function $\Phi(\cdot)$ and state control function $\Psi(\cdot)$.

Let n^i and n^h denote the numbers of the elements in the first and second layer, respectively. The output function used in the MLP is defined as follows:

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{n^h} w^o_{jk} \cdot f(\sum_{i=1}^{n^i} w^h_{ij} x_j + w^h_{0j}) + w^o_{0k}, \tag{13}$$

where $y_k(\cdot)$ represents the k-th component of the function $\boldsymbol{y}(\cdot)$, and $\boldsymbol{x}$ denotes a combined vector of inputs. For instance, $\boldsymbol{x}^T = (\boldsymbol{s}^T, \boldsymbol{u}^T)$ for $\hat{\Phi}(\cdot)$, and $\boldsymbol{x}^T = (\boldsymbol{s}^T, vds^T)$ for $\hat{\Psi}(\cdot)$. w^h denotes the weight coefficients connecting the first layer to the second, and w^o connecting the second layer to the third. w^h_{0j} and w^o_{0k} are bias coefficients. As shown in Fig.5, the activation function $f(\cdot)$ of the elements in the second layer is a differentiable non-linear function, while the activation function of the elements in the first layer and the third is the identity function. We adopted the hyperbolic tangent function $f(\cdot)$ in the second layer as follows:

$$f(v) = \tanh(\frac{v}{\tau}), \tag{14}$$

where τ is a constant value that adjusts nonlinearity and v is the weighted sum of the inputs into the elements.

The parameters of the function w^h_{ij} and w^o_{jk} are modified for each input $\boldsymbol{x}[t]$ to minimize the error $e_s[t]^2$ and $e_u[t]^2$ as defined by Eqn. (8) and (9), using the gradient descent method as follows:

$$\Delta w^h_{ij}[t] = -\eta \frac{\partial}{\partial w^h_{ij}} e[t]^2, \quad \Delta w^o_{jk}[t] = -\eta \frac{\partial}{\partial w^o_{jk}} e[t]^2, \tag{15}$$

where the constant: η denotes a learning rate.

TABLE II
SENSORY STATE AND MOTOR COMMAND.

sensory state	motor command
$\boldsymbol{s}[t] = (s_1[t], s_2[t])$	$\boldsymbol{u}[t] = (u_1[t], u_2[t])$
s_1: horizontal position	u_1: upper-arm roll
s_2: vertical position	u_2: shoulder pitch

III. EXPERIMENT

A. Experimental setting

We performed experiments on sensorimotor learning using the humanoid robot James (Fig.6)[10]. James is a fixed upper-body robotic platform dedicated to vision-based manipulation studies. It is composed of a 7DOF arm with a dexterous 9DOF hand and a 7DOF head as shown in Fig.6. It is equipped with binocular vision, force/torque sensors, tactile sensors, inertial sensors and motor encoders. Low-level sensorimotor information is processed in local control cards, and high-level sensorimotor information is handled in local networks with arbitrary numbers of servers and PCs [12].

The sensory state vector and the motor command vector used in the experiment are presented in Table.II. In this experiment we selected the position in the image of the object of interest, a two dimensional quantity, as the sensory state vector. A sample image obtained from the left eye camera as shown in Fig. 6.

We selected the velocity command for the upper-arm and shoulder joint as the motor command vector. Joint actuation affects the visual position of the object of interest. During exploration, motors were driven stepwise in the following manner: the velocity command was sent to each joint during the first half of the temporal interval δt, and was set to zero during the second half of the interval. We selected a small green marker mounted on the James left arm as the object of interest. The object was detected based on its distinctive color. The color format of the obtained image was transformed from the RGB format to the YUV format to extract the hue robustly. The green regions on the image were then filtered in the YUV domain. The mass centers of the extracted regions were used as the position of the object of interest. Color parameters used to detect green markers were determined experimentally. The color-based region detection was enough robust against external visual noise such as lighting changes and passing people in the visual field. Through the whole experiment, the position of James' head and the eye camera were fixed for simplicity. The sensorimotor system learns only the effects of self-generated arm movements. Therefore, the state prediction and control system are dependent on the orientation of the head.[9]

Experimental parameters are presented in Table III, where E [epoch] denotes the epoch number of the exploration and learning cycle. K and L [ts] (time steps) denote the number of data sampling events and learning events in

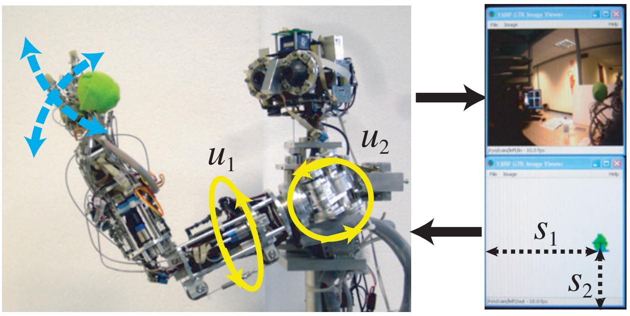


Fig. 6. The humanoid robot James was used for experimental validation of the proposed active sensorimotor learning. Arm position is sensed visually using a green marker mounted on the hand.

TABLE III
EXPERIMENTAL PARAMETERS.

Parameter	Value	Definition
E	20 [epoch]*	exploration-learnig cycle
K	30 [ts]**	exploration iteration
L	10,000 [ts]	learning iteration
δt	2.0 [s]	time step interval
n^i	4	MLP units (1st layer)
n^h	100	MLP units (2nd layer)
n^o	2	MLP units (3rd layer)
η	0.05	MLP learning rate
τ	1.0	MLP parameter
D_w	$[-1.0, +1.0]$	MLP initial weight domain
D_u	$[-0.5, +0.5]$	motor command domain
G	10.0	motor input gain
α	0.1	confidence gain

*epoch: iterated number of the exploration and learning cycle.
**ts: descrete time steps.

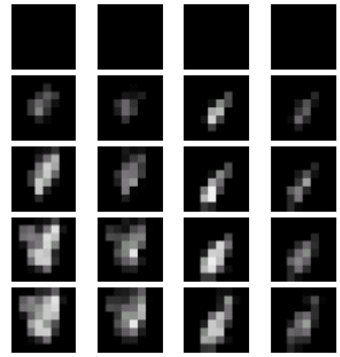


Fig. 7. Temporal evolution of state space confidence. Light intensity indicates the local confidence value. From left to right, the columns correspond to the confidence maps of state prediction in active learning, state control in active learning, state prediction in passive learning, and state control in passive learning, respectively. From top to bottom, the row shows the confidence maps obtained at the end of the 0th, 5th, 10th, 15th, and 20th epoch, respectively.

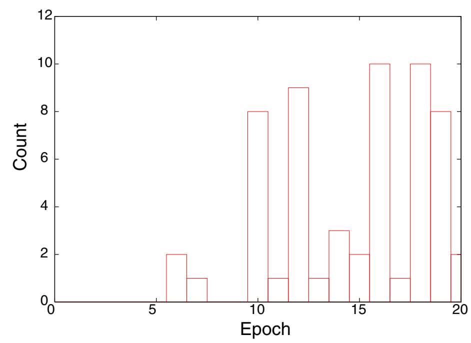


Fig. 8. The number of times that *S-Space* motor command generation was used in each epoch.

each epoch, respectively. The trajectories of the arm were generated randomly in each epoch. All of input and target values for the MLP were normalized in order to better utilize the nonlinearity in the activation function. The initial weight coefficients were randomly selected from the finite domain D_w. The number of hidden elements of the MLP n_h was selected carefully to adjust the function approximating performance of the MLP, to avoid under-fitting or over-fitting problem. Values of the motor commands $u_i[t]$ $(i = 1, 2)$ were randomly selected in domain D_u, proportionally amplified by the gain G, and sent to the motors.

B. Results

We performed both active and passive sensorimotor learning for comparison. Active learning refers to the active motor babbling in *S-Space* and *U-Space* with confidence-based switching. Passive learning refers to the passive motor babbling only in *U-Space*. Fig.7 shows the temporal evolution of state space confidence. In each confidence map, the state space is quantized as 8x8 pixel regions. Light intensity in each region indicates the local confidence value. Each column of Fig.7 contains confidence maps obtained in the same experimental setting. From left to right, the columns correspond to the confidence maps of state prediction in active learning, state control in active learning, state prediction in passive learning, and state control in passive learning, respectively. From top to bottom, each row of the Fig.7 shows confidence maps obtained at the end of the 0th, 5th, 10th, 15th, and 20th epochs. The figure shows that the high-confidence domain in active learning spreads faster than in the case of passive learning, since the exploration by active learning focuses on less well learned states by referring to the confidence value. The active learning strategy avoids learning duplication in the states where learning is complete.

Fig.8 shows the number of times that the *S-Space* was used for motor command generation. For the first several epochs, the *U-Space* was mainly selected, since not enough had been learned to explore the state actively. However, after the 10th epoch, the *S-Space* was mainly selected, since the confidence value reached a sufficient threshold at many state regions.

Fig.9 and Fig.10 show the experimental performance of

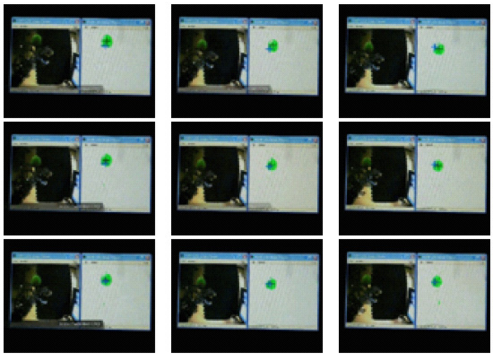


Fig. 9. Three different trials of state prediction. The blue cross indicates the predicted next state.

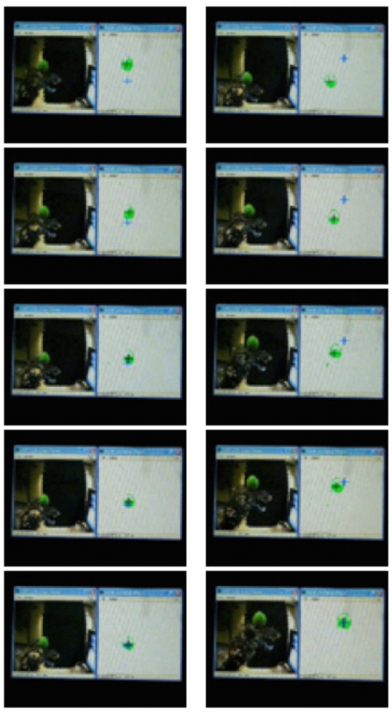


Fig. 10. Two different trials of state control for reaching. The blue cross indicates the target for the state control.

state prediction and state control in the physical space. The results suggest the state prediction and state control worked successfully.

IV. CONCLUSION

Based on a sensorimotor prediction algorithm previously implemented [8], we defined a novel function called the *confidence* function, which works as a memory of reliability for state prediction and control. The aim of this function is to store information about prediction and control reliability for learning, and exploit it for subsequent data sampling. If the robot is sure of its perception and motor behavior, the robot can choose to explore weakly understood areas of interest. This can be used to compensate for reinforcement of important motion primitives. The notion of robotic *confidence* was developed as a first step towards automatically understanding a robot's self and surrounding environment constructively. The approach was discussed theoretically in this paper, and validated in some experiments with a humanoid robot. Although in this experiment the simple case of prediction and control using visual sensing and arm movements was examined, the proposed methodology is not limited to specific modalities and is open for any control approach.

Our global aim is to implement learning as a natural adaptation and self-improvement process for the robot. Accordingly, we must deal with high-dimensional mechanisms to show that our algorithm remains accurate when dealing with complementary sensor data, redundant kinematics, and dynamics. We are now applying the proposed method to the general body recognition. If the robot finds an object which is predictable and controllable, it would be acceptable that robot regards this object as a part of its body. We hope that this direction will lead us to embody a robot's *self-consciousness* using self-generated movements.

ACKNOWLEDGEMENT

Special thanks to Mr. Ivar Thorson of the Italian Institute of Technology for moral support and suggestions.

REFERENCES

[1] J. McCarthy, P. J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, Machine Intelligence 4, pp.463–502, 1969.

[2] M. I. Jordan, D. E. Rumelhart, Forward models: Supervised learning with a distal teacher, Cognitive Science, 16(3), pp.307–354, 1992.

[3] M. Haruno, D. M. Wolpert, M. Kawato, MOSAIC Model for sensorimotor learning and control, Neural Computation, 13, pp.2201–2220, 2001.

[4] M. Kawato, Internal models for motor control and trajectory planning, Current Opinion in Neurobiology, 9, pp. 718–727, 1999.

[5] D. M. Wolpert, Z. Ghahramani, R. J. Flanagan, Perspectives and problems in motor learning, Trends in Cognitive Sciences, 5(11), pp.487–494, 2001.

[6] R. C. Miall, D. M. Wolpert, Forward models for physiological motor control, Neural Networks, 9(8), pp.1265–1279, 1996.

[7] D.M. Wolpert, JR Flanagan, Motor Prediction, Current Biology 11(18) R729-732, 2001

[8] R. Saegusa, F. Nori, G. Sandini, G. Metta, S. Sakka, Sensory prediction for autonomous robots, IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids2007), Pittsburgh, USA, 2007.

[9] R. Saegusa, S. Sakka, G. Metta, G. Sandini, Sensory prediction learning –How to model the self and environment–, The 12th IMEKO TC1-TC7 joint symposium on Man, Science and Measurement, Annecy, France, 2008. (to appear)

[10] L. Jamone, G. Metta, F. Nori and G. Sandini, James, a humanoid robot acting over an unstructured world, Proc. of the Humanoids 2006 conference, pp.143–150, Genoa, Italy, 2006.

[11] D. Rumelhart, J. McClelland, Learning internal representation by error propagation, Parallel Distributed Processing, pp. 318–362, MIT Press, 1984.

[12] G. Metta, P. Fitzpatrick, L. Natale. YARP: Yet another robot platform. International Journal on Advanced Robotics Systems, 3(1):43-48, 2006.