Active Motor Babbling for Sensory-Motor 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, the motor-babbling based sensory-motor learning is considered effective to develop an internal model of the self-body and the environment autonomously. In this paper we propose a methodology of sensory-motor learning and its evaluation towards active learning. The proposed model is characterized by a function called confidence, which works as a memory of reliability for state prediction and control. The confidence for the state can be a good measure to bias the next exploration strategy of data sampling, such as to direct its state to the unreliable domain. We consider the confidence function as the first step to an active behavior design for autonomous environment adaptation. The approach was experimentally validated using the humanoid robot James.**

***Index Terms*—Sensory motor prediction, Neural networks, Learning, humanoid robot, Confidence**

I. Introduction

Learning in robotics is one of the practical solutions 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 too complex to be modeled. Even if the kinematics and the dynamics of the body are known, a real sensory input to the body would be different to one derived from the theoretical model, because the sensory input is always influenced by the interaction with the environment. For instance, when we grasp an object, the physical state of our arm such as a weight and momentum becomes different to those at the normal state. However, 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: in this example, the state of the arm is always different depending on the grasped object. On the other hand, learning provides 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 a 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 the delays and noise in the sensory-motor system. Moreover, motor prediction allows differentiating self-generated movements from externally imposed forces/disturbances [5][6].

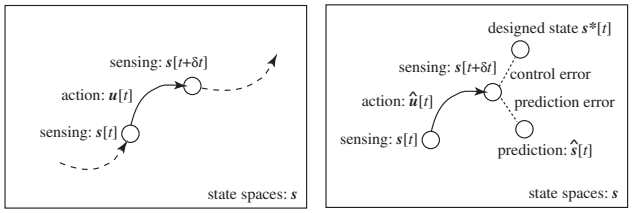
Learning-based perception is applicable not only for motor control but also to model the environment owing to multiple sensorial modalities, such as vision, audition, touch, force/torque, and acceleration sensing. In a similar approach, we developed a learning system aiming at predicting future sensing data based on current sensing data and motor command [8]. Unlike most studies on sensory-motor prediction, the robot and the environment are considered dynamic. Thus, we explored the possibilities for the robot to detect changes in its self or its environment in an autonomous manner: no other information such as a model was given to the system. Following this concept, we investigated a function called *confidence*, driven in the evaluation process of the sensory prediction learning [9]. The aim of this function is to detect inequalities between the predicted situation and the real situation of the body and the environment. The notion of robotic self-confidence was developed as the first step toward self diagnosis and self adaptation.

Our global aim is to implement a learning process as a natural adaptation and self-improvement for the robot. In this context, one of the significant problems in learning is that it requires much time for data sampling and post treatment. An efficient learning strategy is necessary to enhance the learning speed while keeping its quality. The random sampling strategy is considered as the most robust approach for unknown learning environment, on the other hand maybe there are some more formal ways of choosing the sampling strategy

*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
$\boldsymbol{u}$	actuated motor command
$\hat{\boldsymbol{s}}$	estimated next sensory input
$\hat{\boldsymbol{u}}$	estimated motor command
$\boldsymbol{s}^*$	desired next sensory input
$\boldsymbol{u}^*$	desired motor command
$\Phi(\cdot)$	state prediction function
$\Psi(\cdot)$	state transition function
$\chi(\cdot)$	state design function
$\hat{\Phi}(\cdot)$	approximated state prediction function
$\hat{\Psi}(\cdot)$	approximated state transition function

TABLE I
NOTATION OF VARIABLES AND FUNCTIONS.



a) Ideal b) Estimated

Fig. 1. Internal state space of a sensory-motor system. (a) ideal state transition, (b) estimated state transition.

depending on various factors and constrains of the body and the environment, which biases robot learning interest. We propose an improvement of the learning strategy (active motor babbling) based on the confidence for the state, which is an extension of [9] to deal with both of sensory state prediction and control: the current learning evaluation on state prediction and control can be applied to the next exploration strategy of data sampling, which focuses on a state domain of learning interest.

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

II. METHOD

A. Sensory-motor learning

Fig. 1 illustrates internal state space of a sensory-motor system. Notations of the variables used in this figure are defined in the 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 spanned by the state vector. The state is transited by the motor command actuation. Let us consider that the dynamics of $\boldsymbol{s}[t]$ and $\boldsymbol{u}[t]$ can be defined as:

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

Here, for simplicity, a motor command to change the state from $\boldsymbol{s}[t]$ to $\boldsymbol{s}[t+\delta t]$ is assumed as unique based on that δt is a small value. It is a modeling manner of state transition without considering redundancy in the local domain.

The goal of learning is to approximate $\Phi(\cdot)$ and $\Psi(\cdot)$ using data samples acquired through exploration. Let $\hat{\boldsymbol{s}}[t]$ and $\hat{\boldsymbol{u}}[t]$ denote estimated vectors of the next sensory input: $\boldsymbol{s}[t+\delta t]$ and 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:

$$\hat{\boldsymbol{s}}[t] := \hat{\Phi}(\boldsymbol{s}[t], \boldsymbol{u}[t]), \quad (3)$$
$$\hat{\boldsymbol{u}}[t] := \hat{\Psi}(\boldsymbol{s}[t], \hat{\boldsymbol{s}}[t]), \quad (4)$$

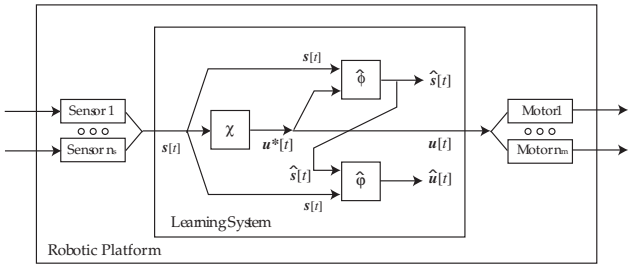
where the estimated next state: $\hat{\boldsymbol{s}}[t]$ is used as an input for the estimation of state transition. The functions $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$ represent internal sensory-motor dynamics, which can be exploited for state prediction and state transition control, as shown in the Fig. 2.

In order to collect learning data for these function approximations, the robot must move its body. In the beginning of learning, however, the robot does not know how to control its joint movement. A motor babbling gives a simple solution for this problem: the learning system randomly generates a motor command $\boldsymbol{u}^*[t_k]$, 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 the motor babbling, the learning system stores measured data: $\{\boldsymbol{s}[t_k], \boldsymbol{u}[t_k], \boldsymbol{s}[t_{k+1}]\}_{k=1,\cdots,K}$ at each time step: t_k. Let us define this motor command generation as the *U-space* motor command generation (Fig.2a). In learning of the eqn. (3) and (4), $\boldsymbol{s}[t_k]$, $\boldsymbol{u}[t_k]$, and $\boldsymbol{s}[t_{k+1}]$ can be used as input vectors of $\boldsymbol{s}[t]$, $\boldsymbol{u}[t]$, and $\hat{\boldsymbol{s}}[t]$, respectively, while $\boldsymbol{s}[t_{k+1}]$ and $\boldsymbol{u}[t_k]$ can be used as target vectors of $\hat{\boldsymbol{s}}[t]$ and $\hat{\boldsymbol{u}}[t]$, respectively.

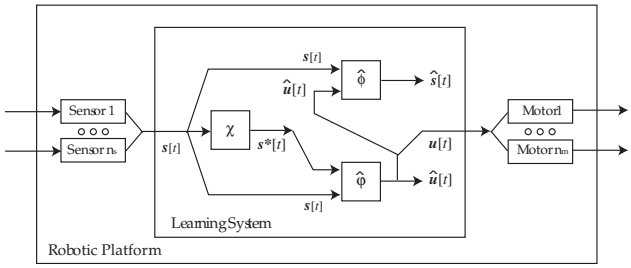
If the learning is performed sufficiently, the robot is able to generate a motor command to reach a desired next state: $\boldsymbol{s}^*[t]$, defined as:

$$\hat{\boldsymbol{s}}[t] = \hat{\Phi}(\boldsymbol{s}[t], \hat{\boldsymbol{u}}[t]), \quad (5)$$
$$\hat{\boldsymbol{u}}[t] = \hat{\Psi}(\boldsymbol{s}[t], \boldsymbol{s}^*[t]), \quad (6)$$

where the estimated motor command: $\hat{\boldsymbol{u}}[t]$ is used as an input for state prediction and actuation of the robot as $\boldsymbol{u}[t] = \hat{\boldsymbol{u}}[t]$, as shown in the Fig. 2b. By using the approximated functions, the robot is able to generate a motor command to collect learning samples of *interest*. Let us define this motor command generation as the *S-space* motor command generation (Fig. 2b).



a) Motor generation in *U-space*



b) Motor generation in *S-space*

Fig. 2. Sensory-motor learning systems.

B. Confidence for a state

Learning results can be evaluated as *confidence* for the considered state. The confidence is based on the state prediction error: $||\boldsymbol{e}_s||$ and motor control error: $||\boldsymbol{e}_u||$ defined as:

$$||\boldsymbol{e}_s[t]|| \quad := \quad ||\hat{\boldsymbol{s}}[t] - \boldsymbol{s}[t]||, \tag{7}$$

$$||\boldsymbol{e}_u[t]|| \quad := \quad ||\hat{\boldsymbol{u}}[t] - \boldsymbol{u}[t]||, \tag{8}$$

where $\hat{\boldsymbol{s}}[t]$ indicates the prediction of $\boldsymbol{s}[t]$ calculated at time $t - \delta t$, and $\hat{\boldsymbol{u}}[t]$ indicates the estimation for motor control calculated at time t, when $\boldsymbol{u}[t]$ $(= \boldsymbol{u}^*[t])$ is given by *U-space* motor command generation. $\boldsymbol{s}[t]$ and $\boldsymbol{u}[t]$ denote the measured sensory input and the actuated motor command at time t, respectively. If the motor command is given by *S-space* motor command generation, Eqn. (8) 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 $\boldsymbol{e}_u[t]$:

$$\boldsymbol{e}_p[t] \quad := \quad \boldsymbol{s}^*[t] - \boldsymbol{s}[t], \tag{9}$$

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 $||\boldsymbol{e}_s||, ||\boldsymbol{e}_u|| \in (0, +\infty)$ into a finite scalar variable: $c[t] \in [0, 1]$ such as

$$c[t] := \exp\left(-\frac{||\boldsymbol{e}_s[t]||^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{||\boldsymbol{e}_u[t]||^2}{2\sigma^2}\right), \tag{10}$$

where the constant: σ^2 determines sensitivity of the filtering (Fig.3). Accumulation of $c[t]$ depending on the state $\boldsymbol{s}[t]$

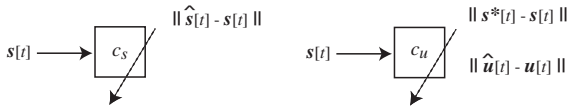


Fig. 3. Confidence of a state for the state prediction function: $\hat{\Phi}$ (left) and the state transition function: $\hat{\Psi}$ (right). Note that the confidence of $\hat{\Psi}$ can be evaluated by two different behavior space: *U-space* or *S-space*.

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:

$$C_{\boldsymbol{s}}[t] := (1 - \alpha)C_{\boldsymbol{s}}[t - \delta t] + \alpha c[t], \tag{11}$$

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. A high value of $C_{\boldsymbol{s}}$ indicates that knowledge of state dynamics at the state $\boldsymbol{s}$ is reliable.

The principal idea of this framing is to exploit confidence derived from the past learning, for the next exploration to collect new learning data of *interest*. If the confidence at the current state is low, for instance, the robot can generate motor babbling in *U-space*. If the confidence is high, the robot can direct its actions in the *S-space* toward the lower confidence state to collect new learning data for improvement of learning, or direct the action toward the state which attracts its attention.

C. Implementation by neural networks

The function approximations of $\hat{\Phi}(\cdot)$ and $\hat{\Psi}(\cdot)$ were implemented with Multi Layer Perceptron (MLP) as shown in Fig.4 [8][11]. MLP is a universal function approximator, which parameters can be optimized by learning. We adopted the MLP with three layers and the conventional gradient descent method as a learning strategy [11].

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

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{n^h} w^o_{jk} \cdot f(\sum_{i=1}^{n^i} w^h_{ij} x_j + w^h_{0j}) + w^o_{0k}, \tag{12}$$

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 $\boldsymbol{x}^T = (\boldsymbol{s}^T, \hat{\boldsymbol{s}}^T)$ for $\hat{\Psi}(\cdot)$. w^h denotes the weight coefficients connecting the first to second layer, and w^o connecting the second to third layer. w^h_{0j} and w^o_{0k} are bias coefficients. As shown in Fig.4, the activation function $f(\cdot)$ of the units in the second layer is a differentiable non-linear function, while the activation functions of the units in the first and the third layers are identity

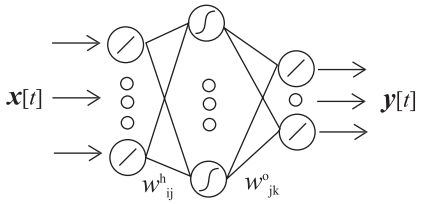


Fig. 4. Multi Layer Perceptron (MLP) usesd for approximation of the state prediction function and state transition function.

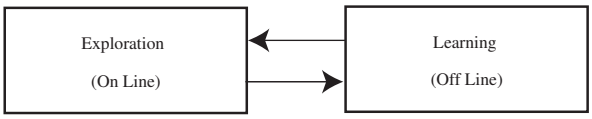


Fig. 5. Learning strategy: a robot explores the environment to collect the next learning data, and evaluates the past learning from on-line experience. After the exploration, the robot optimizes mapping functions with the collected learning samples off-line. These two stages are performed successively. Motor behavior of the robot is generated by motor babbling in *U-space* or *S-space*.

functions. We adopted the hyperbolic tangent function $f(\cdot)$ in the second layer as follows:

$$f(v) = \tanh(\frac{v}{\tau}), \tag{13}$$

where τ is a constant value to control non-linearity and v is the weighted sum of the inputs into the units.

The parameters of the function w^h_{ij} and w^o_{jk} are modified for each input $\boldsymbol{x}[t]$ to minimize the error $||\boldsymbol{e}_s[t]||^2$ and $||\boldsymbol{e}_u[t]||^2$ defined by Eqn. (7) and (8), using the gradient descent method as follows:

$$\Delta w^h_{ij}[t] = -\eta \frac{\partial}{\partial w^h_{ij}} |\boldsymbol{e}[t]|^2, \quad \Delta w^o_{jk}[t] = -\eta \frac{\partial}{\partial w^o_{jk}} |\boldsymbol{e}[t]|^2, \tag{14}$$

where the constant: η denotes a learning rate.

III. Experiment

We show experiments of sensory-motor learning dealing with visual sensing as a sensory input, and arm joints actuation as a motor command effect. When proceeding iterations of on-line joint movements and off-line learning, as illustrated in Fig. 5, the confidence value defined by Eqn. (11) increases as theoretically expected. The behavior of the algorithm will be discussed in this section with the learning results using the humanoid robot James (Fig.6).

A. Learning strategy

The procedure of sensory-motor iterative learning is organized in two stages: *exploration* and *learning*, as illustrated in Fig.5. In the *exploration* stage, the robot generates joints movements randomly (motor babbling) in order to collect learning samples, and evaluates the past learning. In the *learning* stage, on the other hand, the robot optimizes mapping functions with the collected learning samples off-line. Motor behavior of the robot in the *exploration* stage is generated by motor babbling in *U-space* or *S-space*. In this experiment, we set the motor babbling in *U-space* as basic motor behavior. If the confidence of the state at time t is high enough ($C_{\boldsymbol{s}}[t] > \beta$), the robot is allowed to search the minimum-confidence state in the discrete eight-neighbor states, and generate a motor command to direct to this state.

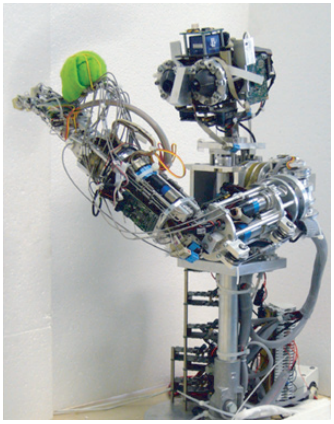


Fig. 6. The humanoid robot James [10].

B. Platform setting

Experiments of the sensory-motor learning were performed using the humanoid robot James (Fig. 6). James is a fixed upper-body robotic platform dedicated to vision-based manipulation studies. It is composed of a seven dof arm with a dexterous nine dof hand and a seven dof head as shown in Fig.6. It is equipped with binocular vision, force/torque, tactile, inertial sensors and encoders. Low-level input and output of sensors and motors are processed in local control cards, and high-level sensory-motor information can be handled in local networks with anonymous numbers of servers and PCs [12].

The selection of the sensory modality and types of motor control are not limited in this framework. In this experiment, we used the image data of the left eye as the sensory input for the sensory-motor learning system, and a velocity command as the system output, which drives actuators of the left arm joints. Motor driving in the experiment was performed stationary: the velocity command is sent to the joints during the first half of the temporal interval δt, while it is set as zero during the second half of the interval. Therefore, James moves and stops at each time step.

The input and output variables are summarized in Table II. The sensory input vector is the horizontal and vertical coordinate of an attention object on an image obtained from the left eye camera, as shown in Fig. 7. The command output vector corresponds to the roll joint of the upper arm and the pitch joint of the shoulder. These joints movements generate the

TABLE II
INPUT AND OUTPUT VARIABLES.

input	output
$\boldsymbol{s}[t] = (x_1[t], x_2[t])$	$\boldsymbol{u}[t] = (u_1[t], u_2[t])$
x_1: horizontal position	u_1: upper-arm roll
x_2: vertical position	u_2: shoulder pitch

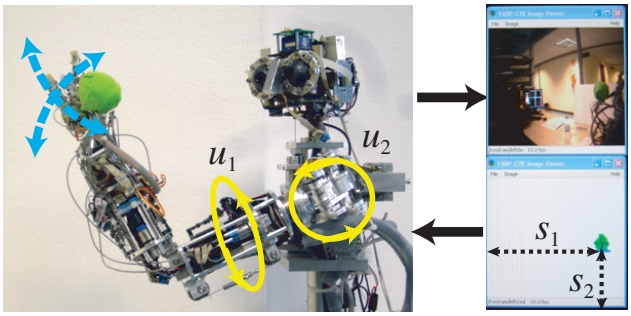


Fig. 7. Experiment setting of sensory-motor learning. The robot moves its arm and the position of the arm is recognized by a green maker mounted on the hand

horizontal and vertical view shift. Let us assume the attention object as a small green maker mounted on the left arm of James. The object is detected based on the color feature. The color format of the obtained image is transformed from the RGB format to the YUV format to extract the hue of color robustly. The green regions on the image are filtered in this domain. The conclusive coordinates of the attention object is the center of the extracted regions. The thresholds for filtering were experimentally determined, and are enough robust to detect the attention object against external visual noise such as lighting change and passing people in the experimental field.

When the robot is sampling data in the exploration stage, the orientation of the head in the task space is fixed for simplicity. Therefore, the prediction system learns the position change of the attention object in the visual field caused only by the self-generated arm movements. Therefore, if the orientation of the head is modified, the state prediction and control are disturbed [9].

C. Parameter setting

The experimental parameters are presented in Table III, where E [epochh] denotes the iterated number of the exploration and learning cycle, K and L [ts] (time steps) denote the number of data sampling and learning in each stage, respectively. The trajectories of the arm were generated at random in each epoch. In order to match domains of input/output values and initial weight coefficients of MLP, all inputs and outputs values for MLP were normalized, and the initial weight coefficients were randomly selected from the finite domain: D_w as described in Table III. The

TABLE III
EXPERIMENTAL PARAMETERS.

Parameter	Value	Definition
E	25 [epoch]*	exploration-learnig cycle
K	50 [ts]**	exploration iteration
L	10,000 [ts]	learning iteration
δt	4.0 [s]	time step interval
n^i	4	MLP units (1st layer)
n^h	30	MLP units (2nd layer)
n^o	2	MLP units (3rd layer)
η	0.01	MLP learning rate
τ	1.0	MLP parameter
D_w	[-0.01, +0.01]	MLP initial weight domain
S_u	$\{-1, 0, +1\}$	motor command set
G	10.0	motor input gain
α	0.1	confidence gain

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

number of the hidden units of MLP: n_h is an design-oriented parameter, which defines the complexity of MLP connected to approximation performance. A value of $u_i[t]$ $(i = 1, 2)$ during motor babbling was randomly selected from a finite set: $S_u = \{-1, 0, +1\}$ for simplicity, whereas the motor command was proportionally amplified by the gain G and sent to the motors.

D. Results

Fig.8 shows maps of state prediction and motor command estimation at the beginning and the late of the exploration-learning epoch: top and bottom figures show the state and motor command maps, and left and right figures show the begining map (at epoch 1) and the late maps (at epoch 20). In the beginning epoch, both plots concentrates in the small domain, while in the late epoch, maps are widely spread to estimate the variables well. Even though the estimation of the motor command are not accurate (see right bottom figure), the system can compensate its analogue outputs by quantizing as {-1, 0, +1} for conclusive outputs. Comparing to the top and bottom figures of the actuated motor command $\boldsymbol{u}$ and the measured state $\boldsymbol{s}$, we can see the nonlinearity from *U-space* to *S-space*, since the square-shape distribution in *U-space* is mapped onto the nonlinearly distorted square-shape distribution in *S-space*. It suggests that even if the robot performs the uniformly random motor babbling in *U-space*, the outcome in *S-space*, where the robot performs tasks, is not uniformly searched. If uniform density of sampling data is required in the *S-space*, the robot must generate motor command in the *S-space*, by using its acquired knowledge of motor control: $\hat{\Psi}(\cdot)$.

Fig.9 shows the evolution of confidence for the state. The left 5x5 images show the state prediction confidence from epoch 1 to epoch 25, while the right 5x5 images show the state transition confidence. Brightness in the image indicates high confidence value (Confidence in [0,1] is mapped onto the intensity in [0,255] proportionally). These two confidence

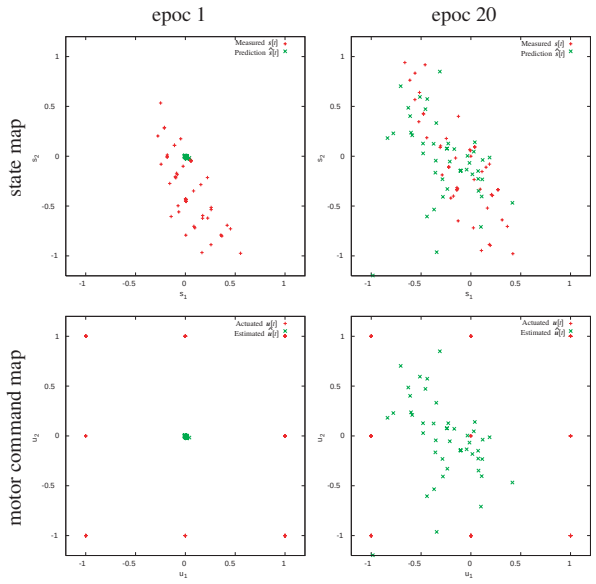


Fig. 8. State maps and motor command maps. Left top: state map at epoch 1, Right top: state map at epoch 20, Left bottom: motor command map at epoch 1, Right bottom: motor command map at epoch 20.

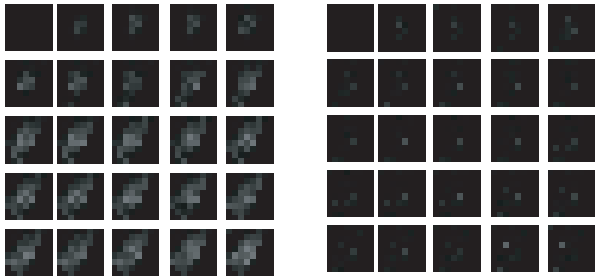


Fig. 9. Temporal sequences of the state prediction confidence (left 5x5 images) and the state transition confidence (right 5x5 images) from epoch 1 to epoch 25. The initial confidence image is at the left top and the final is at the right bottom. The order of image arrangement is like the Z-shape.

Fig. 10. Active data sampling (left: state prediction confidence, right: state transition confidence). Motor command was generated in S-Space to change the current state to the minimum confidence state in eight-neighbors.

maps correspond to the first and second exponential term of Eqn. (10) accumulated by Eqn. (11), respectively. The initial confidence image is located at the left top and the final one is at the right bottom. The order of image arrangement is like the Z-shape. In both image sets, we can see the increase of intensity and area. During the iterations of exploration, the confidence is accumulated. If the target marker is out of view, in the experiment, the joint position is reinitialized at some joint configuration, which allocates the arm near the center of the view. This effect leads to sample data in the center of the view more often than the corner of the view. The confidence, therefore, is greater in the center domain. Comparing to the left and right image sets, the confidence value of the state transition is less than that of the state prediction. One of the reasons for this difference comes from the use of $\hat{\boldsymbol{s}}[t]$ as an input for $\hat{\Psi}(\cdot)$, instead of input $\boldsymbol{s}[t+\delta t]$ which is not available for on-line evaluation at time t, as discussed with Eqn. (4).

Fig.10 shows a motor behavior of active data sampling. The left and right figures show state prediction confidence and state transition confidence at epoch 15, respectively. At the end of this epoch, the current state confidence is detected as a higher value than the preset threshold, and then, motor command was generated in S-Space to change the current state to the minimum confidence state in eight-neighbors (in this case, to the right state indicated by an arrow.) The active data sampling is now in the preliminal level, however, the experimental result suggests its availability. Farther analysis to trace the motor behavior of the active data sampling should be discussed in the next work.

IV. Conclusion

Based on a sensory-motor prediction algorithm previously implemented [8], we defined a novel function called *confidence*, which works as a memory of reliability for state prediction and control. The aim of this function is to store the reliability of learning result for the sensory input, and exploit it for the next data sampling. If the robot is sure of its perception and motor behavior, the robot can decide its exploration and learning based on its learning interest, such as compensation of weak learning part and reinforcement of important motion primitives. The notion of robotic *confidence* was developed as the first step to understand the self and the environment constructively. The approach was discussed theoretically in this paper, and validated positively in some experiments with a humanoid robot. Even if, in the experiment, a simple case of prediction and control using visual sensing and arm movements are examined, the proposed methodology is not limited in some specific modalities and is open for any control approach.

Our global aim is to implement a learning process as a natural adaptation and self-improvement for the robot. We must then deal with the high-dimensional mechanism to show that our algorithm remains accurate when dealing with numerous 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

think that this direction leads us to embody the robot *self-consciousness* by self-generated movements.

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.