

# Active Learning for Sensorimotor Coordinations of Autonomous Robots\*

Ryo Saegusa, Giorgio Metta, Giulio Sandini  
Robotics, Brain and Cognitive Sciences Department  
Italian Institute of Technology

Via Morego 30, 16163 Genoa, Italy

Email: ryos@ieee.org, pasa@liralab.it, giulio.sandini@iit.it

**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. However, learning process requires much time for exploration and computation. In this paper, we propose a method of sensorimotor learning which explores the learning domain actively. Our approach discovers that the embodied learning system can design its own learning process actively, which is different from the conventional passive data-access machine learning. The proposed model is characterized by a function we call the “confidence”, and is a measure of the reliability of state control. The confidence for the state can be a good measure to bias the exploration strategy of data sampling, and to direct its attention to areas of learning interest. We consider the confidence function to be a first step toward an active behavior design for autonomous environment adaptation. The approach was experimentally validated in typical sensorimotor coordination such as arm reaching and object fixation, using the humanoid robot James and the iCub simulator.

**Index Terms**—sensorimotor learning, neural networks, state prediction, state control, 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. Another example is that the sense of touch and force depend on the material of the object and a state of the fingers. 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.

\*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.

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 with the frame problem. An efficient learning strategy is necessary to enhance learning speed while keeping its quality high. A random exploration strategy, such as a conventional motor babbling, is often considered to be the most robust approach for unknown learning domain. Because the random exploration theoretically covers the whole learning domain uniformly. If the learning system has a specific learning domain of interest, however, the random exploration is not a direct way to get learning samples of the domain, and it leads to cost more sampling time. A marked problem

in sensorimotor learning here is that the learning system does not know a correct motor output to reach the domain of interest in the beginning of learning.

We propose an improvement of the exploration strategy: *active motor babbling* based on confidence for the state. That is, the learning evaluation on motor control affects the exploration strategy for learning of motor control. If the evaluation of the motor control is low, the random exploration strategy functions in high possibility. If the evaluation is high, the directed exploration strategy does in high possibility. This approach is an extension of [9] to deal with both the problems of state prediction and state control.

There are some interesting previous studies related to our basic idea [10][11][12][13][14][15][16]. Robbel et al. [10] introduced an active learning approach in motor control, which is based on the LWR and LWPR model [11][13][14] and memory-based resettable motor configuration [12]. These approaches are similar to our approaches. One of our originalities and differences from these approaches is that we do not exactly assume the state recovery (resetability) with a given model such as PID controller. Another difference is that the sensorimotor coordination in our case is visual motor coordination, which is not so trivial like motor-encoder coordination. Active learning of a complex sensorimotor coordination is not yet studied well.

This paper is organized as follows: Section II describes the proposed framework of sensorimotor learning including an introduction of the *confidence* function. Section III describes the experimental results obtained using the humanoid robotic platform James [17] and the iCub robot simulator. 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  $\mathbf{s}[t] \in R^{N_s}$  denote the sensory input vector from the  $N_s$  sensors, and  $\mathbf{u}[t] \in R^{N_m}$  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  $\mathbf{s}[t]$  can be defined as:

$$\mathbf{s}[t + \delta t] = \mathbf{s}[t] + \delta \mathbf{s}[t], \quad (1)$$

$$\delta \mathbf{s}[t] = \Phi^i(\mathbf{s}[t], \mathbf{u}[t]), \quad (2)$$

$$\mathbf{u}[t] = \Psi^i(\mathbf{s}[t], \delta \mathbf{s}[t]). \quad (3)$$

Here, for simplicity, we assume that  $\delta t$  is sufficiently small, and a motor command to change the state from  $\mathbf{s}[t]$  to  $\mathbf{s}[t + \delta t]$  is unique. The goal of learning is to approximate  $\Phi^i(\cdot)$  and  $\Psi^i(\cdot)$  using data samples acquired through exploration. Let  $\delta \hat{\mathbf{s}}[t]$  and  $\hat{\mathbf{u}}[t]$  denote estimated vectors of the sensory input change  $\delta \mathbf{s}[t]$  and that of the actuated motor command  $\mathbf{u}[t]$ ,

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

TABLE I  
VARIABLE AND FUNCTION NOTATION.

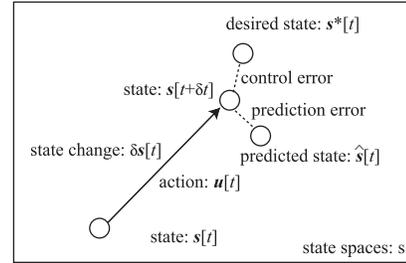


Fig. 1. State transition diagram of the proposed sensorimotor system. Variable notation is defined in TABLE I,

respectively.  $\Phi(\cdot)$  and  $\Psi(\cdot)$  denote the approximations of  $\Phi^i(\cdot)$  and  $\Psi^i(\cdot)$ , defined as:

$$\hat{\mathbf{s}}[t + \delta t] = \mathbf{s}[t] + \delta \hat{\mathbf{s}}[t], \quad (4)$$

$$\delta \hat{\mathbf{s}}[t] = \Phi(\mathbf{s}[t], \mathbf{u}[t]), \quad (5)$$

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

$$\mathbf{u}[t] = \{\mathbf{u}^*[t], \hat{\mathbf{u}}[t]\}, \quad (7)$$

where the desired state change  $\delta \mathbf{s}^*[t]$  is used as an input for the estimation of state control.  $\mathbf{u}[t]$  is selected from a pair of  $\mathbf{u}^*[t]$  and  $\hat{\mathbf{u}}[t]$ . The selection rule is defined later. The diagram of internal state transition is shown in Fig.1.

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 desired motor command  $\mathbf{u}^*[t]$ . The robot then actuates this motor command as  $\mathbf{u}[t] = \mathbf{u}^*[t]$ , leading to a random joint movement. During motor babbling, the learning system stores measured data  $\{\mathbf{s}[t], \mathbf{u}[t], \delta \mathbf{s}[t]\}_{t=t_1, \dots, t_K}$  at each time step  $t$ . Let us refer to the above process as the *U-space* motor command generation (Fig.2). In learning of the functions  $\Phi(\cdot)$  and  $\Psi(\cdot)$ ,  $\mathbf{s}[t]$ ,  $\mathbf{u}[t]$ , and  $\delta \mathbf{s}[t]$  are used as input vectors of these functions, while  $\delta \mathbf{s}[t]$  and  $\mathbf{u}[t]$  are used as target vectors of  $\delta \hat{\mathbf{s}}[t]$  and  $\mathbf{u}^*[t]$ , respectively.

If the learning process is complete, the robot will be able to generate a motor command to reach a desired next state  $\mathbf{s}^*[t]$  by Eqn.6, where the estimated motor command  $\hat{\mathbf{u}}[t]$  is used

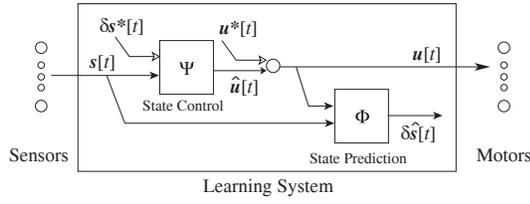


Fig. 2. The proposed learning system to predict and control the state at the next time step. Sources of motor commands are given both in the domain of  $\mathbf{u}$  ( $U$ -Space) or  $\mathbf{s}$  ( $S$ -Space), and one of them are stochastically selected depending on the confidence for the state control.

for actuation of the robot joints as  $\mathbf{u}[t] = \hat{\mathbf{u}}[t]$  represented in Fig. 2. By using the approximated functions in exploration, the robot is able to collect learning samples of *interest* in state space. Let us refer to the above process as the  $S$ -space motor command generation.

Consequently, the robot can refer two manners of motor command generation, such as  $U$ -Space and  $S$ -Space motor generation. The former is useful in the beginning of learning, and the latter is effective to direct the data sampling in the middle phase of learning. We can assume several ways to select or synthesize these two manners. Here, we introduce a simple stochastic synthesizing approach using *confidence*, the temporal evaluation for the state control.

### B. Confidence for a state control

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

$$e_u[t] = |\Psi(\mathbf{s}[t], \delta \mathbf{s}[t]) - \mathbf{u}[t]|. \quad (8)$$

Performance of the motor command generation in  $S$ -space defined as

$$e_p[t] = |\mathbf{s}^*[t - \delta t] - \mathbf{s}[t]|, \quad (9)$$

$$= |\delta \mathbf{s}^*[t - \delta t] - \delta \mathbf{s}[t - \delta t]|. \quad (10)$$

In the following discussion, however, we adopted  $e_u[t]$ .

Let us introduce a modification using the Gaussian function to map  $e_u, e_p \in (0, +\infty)$  onto a finite scalar variable  $c[t] \in [0, 1]$  such as

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

where the constant  $\sigma^2$  determines filtering sensitivity. Accumulation of  $c[t]$  provides robust memory of confidence on state control. Let  $C[t] \in [0, 1]$  denote the *confidence*, working as a temporal moving average of normalized learning error  $c[t]$ . The update rule of the confidence at time  $t$  is defined as:

$$C[t] := (1 - \alpha)C[t - \delta t] + \alpha c[t], \quad (12)$$

where the constant parameter:  $\alpha \in [0, 1]$  denotes an update weight.  $C[0]$  is initialized as zero at the beginning of the learning process. A high value of  $C$  indicates that the state control is reliable. This confidence value is defined at each

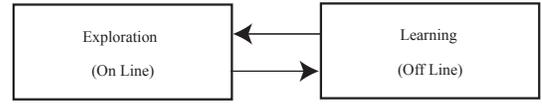


Fig. 3. The proposed learning strategy. A robot explores the environment to collect learning data, and evaluates sensorimotor functions on-line. After exploration, the robot optimizes sensorimotor functions with the collected learning samples off-line. These two processes are repeated alternately until the desired performance is reached.

discretized state of state space, or otherwise simply defined as a representative value of whole state space. In the former state-dependent case, the confidence value of a state is independently updated only when the system receives its state.

### C. Learning strategy

The sensorimotor learning procedure is divided into two stages: *exploration* and *learning*, as illustrated in Fig.3. 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 both in  $U$ -space and  $S$ -space. We used this *confidence* value as a probability to choose a motor command of  $\mathbf{u}$  from a pair of  $\mathbf{u}^*$  and  $\hat{\mathbf{u}}$  (Eqn.7). Therefore, if the confidence for the state control is high, the  $S$ -Space motor generation is selected in high probability.

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 to acquire a specific motor effects such as hand movements in the view. However, we can exploit the advantages of active exploration and learning more dynamically. For instance, we can direct the robot action and learning giving a desired state which attracts its attention depending on the motor skills of the robot.

### D. Implementation by neural networks

The proposed learning system does not limit the type of function approximators for  $\Phi(\cdot)$  and  $\Psi(\cdot)$ . In our implementation, we use Multi Layer Perceptron (MLP) [18], which is known as a universal approximator. The MLP has three layers and is optimized by a simple gradient descent method [18]. The treatments of the MLP in this context is described in the reference [9]. Here, we present a few mathematical formulations and a network structure.

$$y_k(\mathbf{x}) = \sum_{j=1}^{n^h} w_{jk}^o \cdot f\left(\sum_{i=1}^{n^i} w_{ij}^h x_j + w_{0j}^h\right) + w_{0k}^o, \quad (13)$$

where  $y_k(\cdot)$  represents the  $k$ -th component of the function  $\mathbf{y}(\cdot)$ , and  $\mathbf{x}$  denotes a combined vector of inputs.

$$f(v) = \tanh\left(\frac{v}{\tau}\right), \quad (14)$$

TABLE II  
SENSORY STATE AND MOTOR COMMAND FOR JAMES.

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

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

### III. EXPERIMENT

#### A. Experimental setting

We performed experiments on sensorimotor learning using the humanoid robot James [17]. 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.4. 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 inside the body, and high-level sensorimotor information is handled in local networks [19].

The sensory state vector and the motor command vector used in the experiment are presented in Table.II. In this experiment we used the position of the hand of the robot in the image, a two dimensional quantity, as the sensory state vector as shown in Fig. 4. As the motor command vector, we used the position-displacement command of the upper-arm and shoulder joint. Joint actuation affects the visual position of the hand. During exploration, the motor command was sent to each joint at a constant temporal interval of  $\delta t$ . The values of  $\mathbf{u}^*$  were given randomly, while the values of  $\mathbf{s}^*$  were set as a maximumly confident state in the discretized neighbor states of the current state. This implanted motor desire forces the robot to explore the less confident state more positively and skip the states enough learned.

We used a small green marker mounted on the James left arm to recognize the hand. The marker was detected based on its distinctive color with spot noise filtering. The color format of the obtained image was transformed from the RGB format to the YUV format to extract the hue robustly. The mass centers of the extracted green regions were used as the position of interest. Even though color parameters were determined experimentally. The position 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.

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 each epoch, respectively. The trajectories of the arm were generated randomly in each epoch. 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

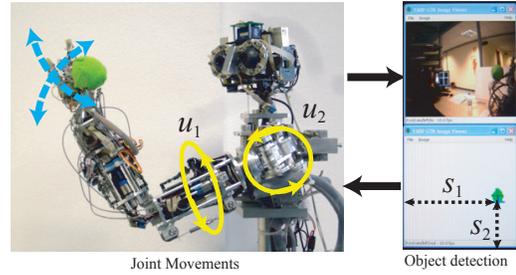


Fig. 4. The humanoid robot James was used for experimental validation of the proposed active sensorimotor learning. Arm position is sensed visually using a green marker mounted on the hand.

TABLE III  
EXPERIMENTAL PARAMETERS.

Parameter	Value	Definition
$E$	20 [epoch]*	exploration-learnig cycle
$K$	30 [ts]**	exploration iteration
$L$	10,000	learning iteration
$\delta t$	1.0 [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)
$\eta$	0.05	MLP learning rate
$\tau$	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
$\alpha$	0.1	confidence gain

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

adjust the function approximating performance of the MLP, to avoid under-fitting or over-fitting problem. Values of the desired motor commands were randomly selected in domain  $D_u$ , proportionally amplified by the gain  $G$ , and sent to the motors.

#### B. Results with James

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 stochastic switching. Passive learning refers to the passive motor babbling only in  $U$ -Space. Fig.5(left) shows the temporal evolution of state space confidence. In each confidence map, the state space is quantized as  $8 \times 8$  pixel regions. Light intensity in each region indicates the local confidence value. 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.5(right) 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. However, after the 10th epoch, the  $S$ -Space was mainly selected, since the confidence value reached a sufficient threshold at many state regions.

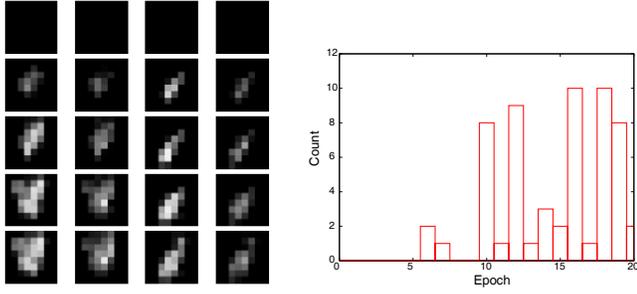


Fig. 5. Analysis of the results. Left: Temporal evolution of state space confidence. Light intensity indicates the local confidence value. From left to right, the columns correspond to the confidence maps of state prediction in active learning, state control in active learning, state prediction in passive 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. Right: The number of times that *S-Space* motor command generation was used in each epoch.



Fig. 6. Three different trials of state prediction. The blue cross indicates the predicted next state.

Fig.6 and Fig.7 show the experimental performance of state prediction and state control in the physical space. The results suggest the state prediction and state control worked successfully.

### C. Results with iCub Simulator

We performed farther experiments with the iCub simulator (Fig.8), which is an ODE-based robot simulator designed for the humanoid robot, iCub [19]. The aim of this experiment is to examine how the proposed learning approach functions in higher-dimensional state space. We performed simultaneous two learning processes of body movements; the object fixation and the arm reaching. Regarding the object fixation, the state was set as four-dimensional input composed of the horizontal and vertical coordinate of its own hand in the left and right view. The motor action was corresponding to three cooperated movements of the both eyes; horizontal, vertical and vergence movements (TABLE IV, top). Regarding the arm reaching, the state was set as three-dimensional input composed of the horizontal, vertical and vergence position of both eyes. The motor action was corresponding to the upper-arm roll, shoulder pitch and elbow pitch, which enables three-dimensional hand movements (TABLE IV, bottom). The motor actions of the



Fig. 7. Two different trials of state control for reaching. The blue cross indicates the target for the state control.

TABLE IV  
SENSORY STATE AND MOTOR COMMAND FOR THE ICUB SIMULATOR.

sensory state(fixation)	motor command(fixation)
$\mathbf{s}[t] = (s_1[t], s_2[t], s_3[t], s_4[t])$	$\mathbf{u}[t] = (u_1[t], u_2[t], u_3[t])$
$s_1$ : horizontal position (L)	$u_1$ : horizontal eye roll
$s_2$ : vertical position (L)	$u_2$ : vertical eye roll
$s_3$ : horizontal position (R)	$u_3$ : vergence eye roll
$s_4$ : vertical position (R)	
sensory state(reaching)	motor command(reaching)
$\mathbf{s}[t] = (s_1[t], s_2[t], s_3[t])$	$\mathbf{u}[t] = (u_1[t], u_2[t], u_3[t])$
$s_1$ : horizontal eye position	$u_1$ : upper-arm roll
$s_2$ : vertical eye position	$u_2$ : shoulder pitch
$s_3$ : vergence eye position	$u_3$ : elbow pitch

object fixation and arm reaching are independent, however the state of the arm reaching depends on the motor action of the object fixation.

The learning process of the object fixation was run firstly, followed by the learning of the arm reaching. The task goal of the object fixation is to move the both eyes to watch the hand in the center of the left and right view. The task goal of the arm reaching is to move its hand to the desired position in the view. The intervals of the two actions were set up differently. The interval of the eye movement was set up five times faster than the interval of the arm movement. Therefore, the hand was normally caught in the center of the view by fixation, and then, the robot reaches out the arm from the center of the view to the desired position. The dimension of the two learning processes was totally seven for the state and six for motor action. Therefore, state prediction and control requires 13-7 dimensional input-output mapping and 14-6 dimensional input-output mapping, respectively. For simplicity, the representative state confidence was used in this learning. In the same manner as the previous experiments with James, the values of  $\mathbf{u}^*$  was given randomly in the both



Fig. 8. The iCub Simulator (left) and the binocular visual processing system (right).

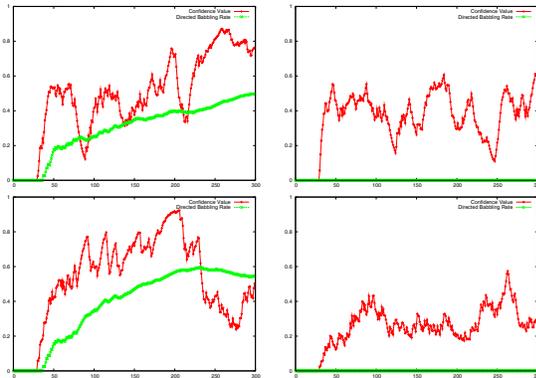


Fig. 9. The temporal evolution of confidence value and the activeness rate with the iCub Simulator. Fixation in the active mode (top left), fixation in the random mode (top right), arm reaching in the active mode (bottom left), arm reaching in the random mode (bottom right)

learning of fixation and reaching. For fixation learning,  $s^*$  was set as the center of the view (0,0,0). For learning of reaching, the values of  $s^*$  was given randomly to explore the view field (state space) uniformly, canceling biased S-space exploration nonlinearly-projected by U-space uniform exploration.

Both learning processes were performed successfully in terms of learning acceleration by the proposed active learning approaches. After learning, the robot in the simulator successfully fixated its hand and moved it as desired in the view field. Here we just present some plots in Fig. 9. These figures shows temporal evolution of the confidence values and activeness rates regarding the object fixation and arm reaching. The results suggested that active learning was helpful to master sensorimotor skills faster than the normal random (passive) learning in this high-dimensional learning cases.

#### IV. CONCLUSION

Based on a sensorimotor prediction algorithm previously implemented [9], 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 reliability of state control, and exploit it for subsequent data sampling. If the robot is sure of its motor behavior, the robot can direct the exploration in 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 of 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 the experiments examined the simple cases of sensorimotor coordination, the proposed framework is not limited to some specific modalities and is open for any sensorimotor setting.

Our global aim is to implement learning as a natural adaptation and self-improvement process for the robot. Accordingly, we must deal with more high-dimensional mechanisms to show that our algorithm remains accurate when dealing with complementary sensor data, redundant kinematics, and dynamics. We are also applying the proposed method to the general robot coaching, expecting that this direction will lead us to embody a robot's *interactive education*.

#### REFERENCES

- [1] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," *Machine Intelligence*, no. 4, pp. 463–502, 1969.
- [2] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, no. 3, pp. 307–354, 1992.
- [3] M. Haruno, D. M. Wolpert, and M. Kawato, "Mosaic model for sensorimotor learning and control," *Neural Computation*, no. 13, pp. 2201–2220, 2001.
- [4] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, no. 9, pp. 718–727, 1999.
- [5] D. M. Wolpert, Z. Ghahramani, and F. R. J., "Perspectives and problems in motor learning," *Trends in Cognitive Sciences*, vol. 5, no. 11, pp. 487–494, 2001.
- [6] R. C. Miall and D. M. Wolpert, "Forward models for physiological motor control," *Neural Networks*, vol. 9, no. 8, pp. 1265–1279, 1996.
- [7] D. Wolpert and J. Flanagan, "Motor prediction," *Current Biology*, vol. 11, no. 18, pp. R729–732, 2001.
- [8] R. Saegusa, F. Nori, G. Sandini, G. Metta, and S. Sakka, "Sensory prediction for autonomous robots," in *IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids2007)*, 29th November - 1st December 2007.
- [9] R. Saegusa, S. Sakka, G. Metta, and G. Sandini, "Sensory prediction learning - how to model the self and environment -," in *Proceedings of the 12th IMEKO TC1-TC7 joint Symposium on "Man Science and Measurement" (IMEKO2008)*, September 3rd-5th 2008, pp. 269–275.
- [10] P. Robbel, "Active learning in motor control," Ph.D. dissertation, School of Informatics, University of Edinburgh, 2005.
- [11] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, June 2005.
- [12] S. Schaal and C. G. Atkeson, "Robot juggling: An implementation of memory-based learning," *Control Systems Magazine*, vol. 14, no. 1, pp. 57–71, 1994.
- [13] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11, pp. 11–73, 1997.
- [14] —, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, pp. 75–113, 1997.
- [15] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.
- [16] S. Thrun, "The role of exploration in learning control," in *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold, 1992.
- [17] L. Jamone, G. Metta, F. Nori, and G. Sandini, "James: A humanoid robot acting over an unstructured world," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 4–6 Dec. 2006, pp. 143–150.
- [18] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*. MIT Press, 1984, ch. Learning internal representation by error propagation, pp. pp.318–362.
- [19] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.