Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

A Probabilistic Model of MOSAIC

Satoshi Osaga, Jun-ichiro Hirayama, Takashi Takenouchi and Shin Ishii
Graduate School of Information Science
Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara, 630-0192, Japan

Abstract

Humans can generate accurate and appropriate motor commands in various and even uncertain environments. MOSAIC (MOdular Sellection And Identification for Control) was formerly proposed for describing such human ability, but it includes some complex and heuristic procedures which make the model's understandability hard. In this article, we present an alternative and probabilistic model of MOSAIC (p-MOSAIC) as a mixture of normal distributions, and an online EM-based learning method for its predictors and controllers. Theoretical consideration shows that the learning rule of p-MOSAIC corresponds to that of MOSAIC except for some points mostly related to the controller learning. Experimental studies using synthetic datasets have shown some practical advantages of p-MOSAIC. One is that the learning rule of p-MOSAIC makes the estimation of 'responsibility' stable. Another is that p-MOSAIC realizes accurate control and robust parameter learning in comparison to the original MOSAIC especially in noisy environments, due to the direct incorporation of the noise into the model.

1 Introduction

Humans have the remarkable ability to generate accurate and appropriate motor commands in various and even uncertain environments. Studies of human motor controls have shown that dis-adaptation and re-adaptation to a learned environment are more rapid than adaptation to a novel environment [7], implying that the human motor control could be performed by a modular structure consisting of multiple controllers each adapting to a specific environment.

MOSAIC [2] was formerly proposed for modeling the motor control system with such a modular structure. In MOSAIC, each controller is coupled with a corresponding predictor, and a motor command is determined by a weighted mean of outputs of multiple controllers, where the weight for each controller (responsibility) is estimated based on the prediction error of the corresponding predictor. However, the current MOSAIC includes some complex and heuristic procedures which make the model's understandability hard.

In this study, we re-formulate the MOSAIC as a probabilistic model in order to make an easily understandable framework. Parameters of predictors and controllers are estimated by the online EM algorithm [3], which maximizes the log-likelihood of the model, given the history of control results. We also show results of computer simulations in which behaviors of responsibility and controller learning of p-MOSAIC are compared with those of MOSAIC.

2 MOSAIC

We consider a situation where the dynamics of the motor system is given by a discrete-time system:

$$\tilde{x}_{t+1} = \Phi(\tilde{x}_t, u_t), \tag{1}$$

where $\tilde{x}_t$ and u_t are the system state and the applied motor command, respectively, at time t. The task of the motor control is to make the system state $\tilde{x}_t$ to keep on a given trajectory x_t^*.

To perform this control task, we assume M pairs of a controller and a predictor. The objective of the controller is to generate an appropriate motor command u_t which well produces the desired state x_{t+1}^*. We assume that an output of the i-th controller is represented as

$$\psi_{i,t} = \psi(\tilde{x}_t, x_{t+1}^*; v_i), \tag{2}$$

where v_i is the variable parameter of the i-th controller. On the other hand, the objective of the predictor is to well predict the system state at the next time step, then an output of the i-th predictor is given by

$$\phi_{i,t} = \phi(\tilde{x}_{t-1}, u_{t-1}; w_i), \tag{3}$$

where w_i is the variable parameter of the i-th predictor. Because there are M pairs of a controller and a predictor, the responsibility for each controller (and

1-4244-0703-6/07/$20.00

predictor) should be defined. The responsibility signal $\lambda_{i,t}$ for the i-th pair is defined by

$$\lambda_{i,t} = \frac{\exp(-|\tilde{x}_t - \phi_{i,t}|^2/\sigma^2)\hat{\lambda}_{i,t}}{\sum_{j=1}^{M}\exp(-|\tilde{x}_t - \phi_{j,t}|^2/\sigma^2)\hat{\lambda}_{j,t}}, \tag{4}$$

where σ is a constant and $\hat{\lambda}_{i,t}$ is a rough prediction of the responsibility signal $\lambda_{i,t}$ which is typically given as a constant (then ignored). The responsibility represents how well each predictor reproduces the target dynamics, and then, an overall motor command $\tilde{u}_t$ at time t is given by a linear combination of outputs $\psi_{i,t}$ of the M controllers as

$$\tilde{u}_t = \sum_{i=1}^{M} \lambda_{i,t}\psi_{i,t} + u_t^{\text{fb}}. \tag{5}$$

Here, u_t^{fb} is a feedback motor command, which is assumed to be produced by a PID or PAD controller, based on the difference between x_t^* and $\tilde{x}_t$.

MOSAIC is trained by updating the parameters of controllers and predictors. A learning rule is given by

$$\Delta v_i = \kappa\lambda_{i,t}\frac{\partial\psi_{i,t}}{\partial v_i}(u_t^* - \psi_{i,t}) \tag{6}$$

$$\Delta w_i = \kappa\lambda_{i,t}\frac{\partial\phi_{i,t}}{\partial w_i}(\tilde{x}_t - \phi_{i,t}), \tag{7}$$

where Δv_i and Δw_i are the updates of parameters v_i, w_i in a single learning step, that is,

$$v_i^{(t)} = v_i^{(t-1)} + \Delta v_i$$
$$w_i^{(t)} = w_i^{(t-1)} + \Delta w_i,$$

where the subscripts, $t-1, t$, mean the time steps, $t-1, t$, respectively, κ is the learning rate, and u_t^* is the desired motor command. Although it is assumed that the desired motor command u_t^* is available in Eq. (6), this assumption is not practical. Then, the controller learning (6) is approximately performed using the feedback-error learning [6] as

$$\Delta v_i \approx \kappa\lambda_{i,t}\frac{\partial\psi_{i,t}}{\partial v_i}u_t^{\text{fb}}. \tag{8}$$

3 p-MOSAIC

With a set of M predictors, $\tilde{x}_t = \phi(\tilde{x}_{t-1}, u; w_i) + \varepsilon_i$, where ε_i is the noise of the i-th predictor, the state prediction by integrating those predictions is given probabilistically as a mixture of normal distributions:

$$\begin{aligned} &p(x_t|\tilde{x}_{t-1}, u_{t-1}; \boldsymbol{\lambda}, \boldsymbol{w}) \\ =&\sum_{i=1}^{M}\lambda_i \mathrm{N}(x_t|\phi(\tilde{x}_{t-1}, u_{t-1}; w_i), \alpha_i^{-1}) \end{aligned} \tag{9}$$

where x_t is a random variable for the predicted state at time t, $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_M)$ is the mixing rate vector such that $\lambda_i \geq 0$ and $\sum_{i=1}^{M}\lambda_i = 1$, and $\boldsymbol{w} = (w_1, \cdots, w_M)$ is the set of predictors' parameters. In our particular experiments in section 4, we use a linear predictor:

$$\phi(x_{t-1}, u_{t-1}; w_i) = w_{i,x}x_{t-1} + w_{i,u}u_{t-1}. \tag{10}$$

On the other hand, the motor command is deterministically given by Eq. (5):

$$\begin{aligned} &p(u_{t-1}|\tilde{x}_{t-1}, x_t^*; \boldsymbol{\lambda}, \boldsymbol{v}) \\ =&\delta(u_{t-1} - \sum_{i=1}^{M}\lambda_i\psi(\tilde{x}_{t-1}, x_t^*; v_i) - u_{t-1}^{\text{fb}}) \\ =&\delta(u_{t-1} - \tilde{u}_{t-1}), \end{aligned} \tag{11}$$

where $\boldsymbol{v} = (v_1, \cdots, v_M)$ is the set of controller parameters and $\delta(\cdot)$ is Dirac's delta function. Then, the probability of being state x_t, given the previous state $\tilde{x}_{t-1}$ and the desired state x_t^*, is obtained by marginalizing the motor command u_{t-1} as

$$\begin{aligned} &p(x_t|\tilde{x}_{t-1}, x_t^*; \boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{v}) \\ =&\int p(x_t|\tilde{x}_{t-1}, u_{t-1}; \boldsymbol{w})p(u_{t-1}|\tilde{x}_{t-1}, x_t^*; \boldsymbol{v})du_{t-1} \\ =&\sum_{i=1}^{M}\lambda_i\mathrm{N}(x_t|\phi(\tilde{x}_{t-1}, \tilde{u}_{t-1}; w_i), \alpha_i^{-1}). \end{aligned} \tag{12}$$

For a desired trajectory $x_{1:T}^* = (x_1^*, \cdots, x_T^*)$ and an actual trajectory $\tilde{x}_{0:T} = (\tilde{x}_0, \cdots, \tilde{x}_T)$, the probability of a state sequence $x_{1:T} = (x_1, \cdots, x_T)$ of random variables is represented as

$$p(x_{1:T}|\tilde{x}_{0:T}, x_{1:T}^*; \boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{v}) = \prod_{t=1}^{T} p(x_t|\tilde{x}_{t-1}, x_t^*; \boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{v}), \tag{13}$$

where the random variables are assumed to be independent of each other (see Fig. 1). Given $x_{1:T}^*$ and $\tilde{x}_{1:T}$, the parameters of the predictors and the controllers are determined by the maximum likelihood estimation. In the following two subsections, we describe learning rules of the predictors and the controllers.

3.1 Learning rule of predictors

Parameters $\boldsymbol{\lambda}$ and $\boldsymbol{w}$ of the predictors are primarily estimated so as to maximize the log-likelihood:

$$\sum_{t=1}^{T}\log p(x_t = \tilde{x}_t|\tilde{x}_{t-1}, x_t^*; \boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{v}), \tag{14}$$

by means of the online EM algorithm, in which the controller parameters $\boldsymbol{v}$ are fixed. By introducing a

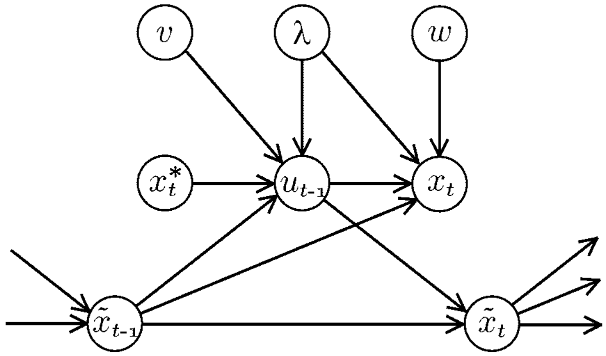

Figure 1: Graphical model [5] of p-MOSAIC.

hidden variable c_t which indexes predictor-controller pairs, the online free energy for any distribution of the hidden variable, $q_{\mathrm{p}}(c_t)$, is defined as

$$\begin{aligned} & F_T[\{q_{\mathrm{p}}(c_t)\}, \boldsymbol{\lambda}, \boldsymbol{w}] \\ =& \sum_{t=1}^{T} \Gamma_T(t) \left\langle \log \frac{q_{\mathrm{p}}(c_t)}{p(\tilde{x}_t, c_t | \tilde{x}_{t-1}, x_t^*, \boldsymbol{\lambda}, \boldsymbol{w})} \right\rangle_{q_{\mathrm{p}}(c_t)}, \end{aligned} \tag{15}$$

where $p(\tilde{x}_t, c_t|\tilde{x}_{t-1}, x_t^*, \boldsymbol{\lambda}, \boldsymbol{w}) = \mathrm{N}(\tilde{x}_t|\phi_{c_t,t}, \alpha_{c_t}^{-1})\lambda_{c_t}$. $< \cdot >_{q_{\mathrm{p}}(c_t)}$ is the expectation with respect to the distribution $q_{\mathrm{p}}(c_t)$, and $\Gamma_T(t)$ is given by

$$\Gamma_T(t) = \begin{cases} 1 & (t = T) \\ \prod_{s=t+1}^{T} \gamma_s & (0 \le t < T), \end{cases}$$

where γ_s $(0 \le \gamma_s < 1)$ is called the forgetting factor. The online free energy is minimized according to the online EM algorithm, in which the following two steps are alternately repeated after seeing x_T^* and $\tilde{x}_{T-1}$ at a time step T:

E-step

$$\begin{aligned} q_{\mathrm{p}}(c_T) &= \frac{p(\tilde{x}_T, c_T|\tilde{x}_{T-1}, x_T^*, \boldsymbol{\lambda}, \boldsymbol{w})}{\sum_{i=1}^{M} p(\tilde{x}_T, c_T = i|\tilde{x}_{T-1}, x_T^*, \boldsymbol{\lambda}, \boldsymbol{w})} \\ &= \frac{\mathrm{N}(\tilde{x}_t|\phi_{c_T,T}^{(T-1)}, 1/\alpha_{c_T}^{(T-1)})\lambda_{c_T}^{(T-1)}}{\sum_{i=1}^{M} \mathrm{N}(\tilde{x}_T|\phi_{i,T}^{(T-1)}, 1/\alpha_i^{(T-1)})\lambda_i^{(T-1)}}. \end{aligned} \tag{16}$$

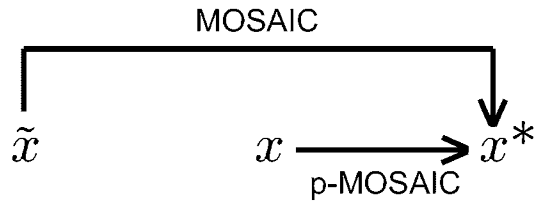

Figure 2: Difference of learning method of controller.

M-step

$$\lambda_i^{(T)} = (1-\eta_T)\lambda_i^{(T-1)} + \eta_T q_{\mathrm{p}}(c_T = i), \tag{17}$$

$$\begin{aligned} \Delta w_i^{(T)} &= (1-\eta_T)\Delta w_i^{(T-1)} \\ &\quad + \eta_T \kappa \alpha_i q_{\mathrm{p}}(c_T = i)(\tilde{x}_T - \phi_{i,T})\frac{\partial \phi_{i,T}}{\partial w_i}, \end{aligned} \tag{18}$$

$$\alpha_i^{(T)} = \left((1-\eta_T)/\alpha_i^{(T-1)} + \eta_T(\tilde{x}_T - \phi_{i,T})^2\right)^{-1}, \tag{19}$$

where η_T is given by

$$\begin{aligned} \eta_T &= 1/N_T, \\ N_T &= \gamma_T N_{T-1} + 1 \quad (N_0 = 0). \end{aligned}$$

Although the above learning rules of p-MOSAIC involve a smoothing effect on the sufficient statistics in the M-step, due to the online free energy, they become similar to the learning rules of MOSAIC in a special setting of $\gamma_t = 0 (t = 1, \cdots, T)$, which corresponds to discarding the smoothing effect. Even in this special setting, however, the learning rule of p-MOSAIC contains an additional term associated with the inverse variance α_i of each predictor (Eq. (18)), which represents the noise level of the predictor.

3.2 Learning method of controllers

The controller parameters $\boldsymbol{v}$ are primarily estimated so as to maximize the log-likelihood:

$$\sum_{t=1}^{T} \log p(x_t = x_t^* | \tilde{x}_{t-1}, x_t^*; \boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{v}),$$

while the predictor parameters, $\boldsymbol{\lambda}$ and $\boldsymbol{w}$, are fixed. According to the online EM algorithm, instead of the log-likelihood, the online free energy:

$$\begin{aligned} & F_T[\{q_{\mathrm{c}}(c_t)\}, \boldsymbol{v}] \\ =& \sum_{t=1}^{T} \Gamma_T(t) \left\langle \log \frac{q_{\mathrm{c}}(c_t)}{p(x_t = x_t^*, c_t | \tilde{x}_{t-1}, x_t^*, \boldsymbol{v})} \right\rangle_{q_{\mathrm{c}}(c_t)} \end{aligned} \tag{20}$$

for any distribution of the hidden variable, $q_c(c_t)$, is minimized, where $p(x_t = x_t^*, c_t|\tilde{x}_{t-1}, x_t^*, \boldsymbol{v}) = \mathrm{N}(x_t^*|\phi_{c_t,t}, \alpha_{c_t}^{-1})\lambda_{c_t}$. $< \cdot >_{q_c(c_t)}$ is the expectation with respect to the distribution $q_c(c_t)$. As an incremental minimization of the online free energy, the following two steps are alternately repeated, given the desired state x_T^* and the previous state $\tilde{x}_{T-1}$:

E-step

$$q_c(c_T) = \frac{p(x_T = x_T^*, c_T|\tilde{x}_{T-1}, x_T^*, \boldsymbol{v})}{\sum_{i=1}^{M} p(x_T = x_T^*, c_T = i|\tilde{x}_{T-1}, x_T^*, \boldsymbol{v})} = \frac{\mathrm{N}(x_T^*|\phi_{c_T,T}^{(T-1)}, 1/\alpha_{c_T}^{(T-1)})\lambda_{c_T}^{(T-1)}}{\sum_{i=1}^{M} \mathrm{N}(x_T^*|\phi_{i,T}^{(T-1)}, 1/\alpha_i^{(T-1)})\lambda_i^{(T-1)}}. \quad (21)$$

M-step

$$\Delta v_i^{(T)} = (1-\eta_T)\Delta v_i^{(T-1)} + \eta_T \kappa \lambda_i \frac{\partial \psi_{i,T-1}}{\partial v_i} \sum_{j=1}^{M} \alpha_j q(c_T = j) w_{j,u}(x_T^* - \phi_{j,T}). \quad (22)$$

Here, $w_{j,u}$ is the predictor parameter defined in Eq. (10). Even if the forgetting factor γ_t is constant at zero, the M-step equation reduces to

$$\Delta v_i^{(T)} = \kappa \lambda_i \frac{\partial \psi_{i,T-1}}{\partial v_i} \sum_{j=1}^{M} \alpha_j q(c_T = j) w_{j,u}(x_T^* - \phi_{j,T}), \quad (23)$$

which is obviously different from Eq. (6), the learning rule of controllers in MOSAIC. The controller learning in MOSAIC is defined as a gradient-based feedback-error learning, which tries to minimize the time-lag difference between the previous actual state $\tilde{x}_{t-1}$ and the previous desired state x_{t-1}^*. On the other hand, the controller learning in p-MOSAIC tries to minimize the difference between the current predicted state $\hat{x}_t$ and the current desired state x_t^*. Moreover, the learning rule of p-MOSAIC includes the inverse variance α_j (Eq. (22)). These two points come from the difference in the learning criteria in MOSAIC and p-MOSAIC (see Fig. 2).

4 Simulation studies

To compare p-MOSAIC with MOSAIC, we simulated the control of a spring-mass-damper system as depicted in Fig. 3. Let Fig. 4 show the desired trajectory of the object (mass position) for 12 seconds. To see the adaptability of the motor control system, three different environments (difference in mass of the object, damping and spring constants, see Table 1) switch every 4 seconds. In both of MOSAIC and p-MOSAIC, we prepared three predictor-controller pairs. The observation and control were done with 1000 Hz, and a single trial was continued for 12 seconds. The predictors (10) were input by the motor command, the state (position and velocity) of the object at the present time, and output the predicted acceleration of the object at the next time. The controllers were input by the state at the present time and the desired acceleration at the next time, and output a motor command at the present time. In this simulation, we used a PAD controller to produce the feedback motor command. Note that our task for the spring-mass-damper system is almost the same as in the previous work [4]. A regularization term was introduced to the estimation of responsibility in MOSAIC and p-MOSAIC, in order to suppress any overfitting to the noisy environment [1].

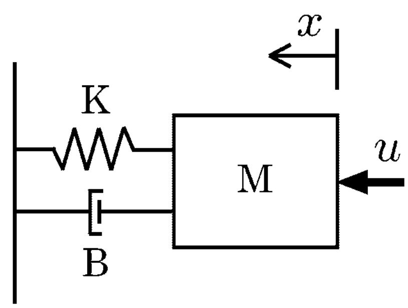

Figure 3: A spring-mass-damper system.

	M(Kg)	B(Nm^{-1}s)	K(Nm^{-1})	NoiseVar
1	1.0	2.0	8.0	0.1 or 1.0
2	5.0	7.0	4.0	0.1 or 1.0
3	8.0	3.0	1.0	0.1 or 1.0

Table 1: The properties of the three environments.

4.1 Responsibility

We first examined how the responsibility behaves. Before the experiment, three predictor-controller pairs were completely trained to adapt individually to their own environments in Table 1. Since there is no learning factor, we can compare solely the estimation of the responsibility between Eq. (17) with the forgetting factor being zero (for comparison), and Eq. (4).

Fig. 5 shows the result. Although p-MOSAIC achieved a complete switching of controllers in re-

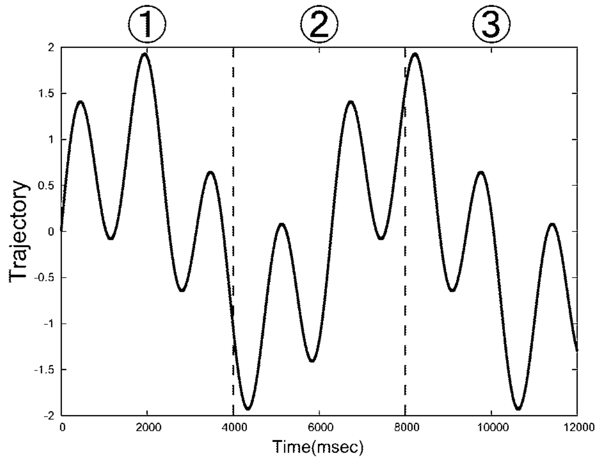

Figure 4: A desired trajectory.

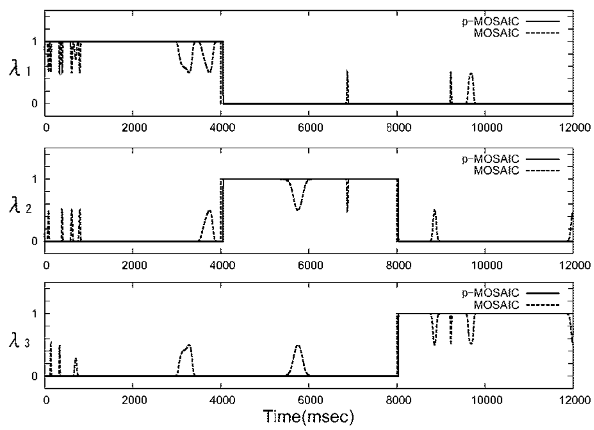

Figure 5: The responsibility along time.

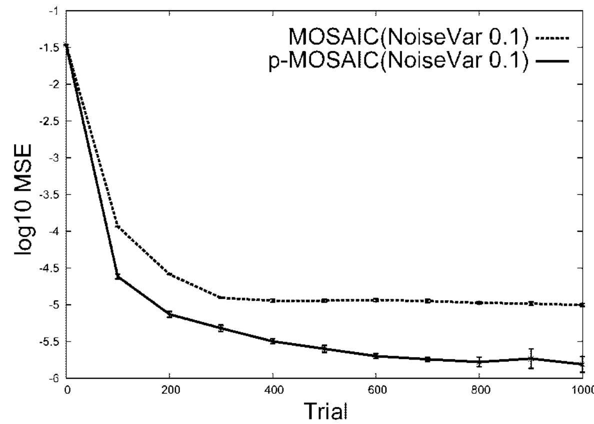

Figure 6: The logarithm of mean square error between the actual trajectory and the desired trajectory against trials.

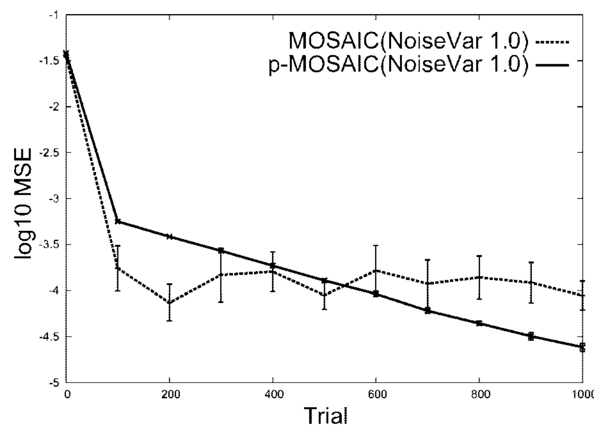

Figure 7: The logarithm of mean square error against trials.

sponse to environmental changes, MOSAIC sometimes failed.

4.2 Controller learning

Next, we compared the controller learning, Eq. (22) of p-MOSAIC, and Eq. (6) of MOSAIC, assuming predictors were completely trained to adapt to their own environments. For comparison of controller learning solely, we used Eq. (17) in both MOSAIC and p-MOSAIC for estimation of responsibility, and the forgetting factor was fixed at zero. We examined the controller learning in particular when the actual state $\tilde{x}_t$ is disturbed by a noise.

Figures 6 and 7 show the results for cases with a small noise and a relatively large noise, respectively. When the noise level is low (Fig. 6), p-MOSAIC achieved more accurate control than MOSAIC. When the noise level is relatively high (Fig. 7), the learning by MOSAIC proceeded faster, but it was substantially unstable; hence, the performance became better by p-MOSAIC after about 1000 trials. In the early learning phase, the controller learning of p-MOSAIC proceeded slowly due to the control of the inverse variance α_i. Because the noise of the environment is large, the adaptive control of the inverse variance made the learning slow but stable, suggesting the effectiveness of adaptive adjustment of learning speed in p-MOSAIC.

4.3 Stability of learning

In the previous subsection, we assumed that each predictor is completely adapted to each environment and focused on the controller learning. In this sub-

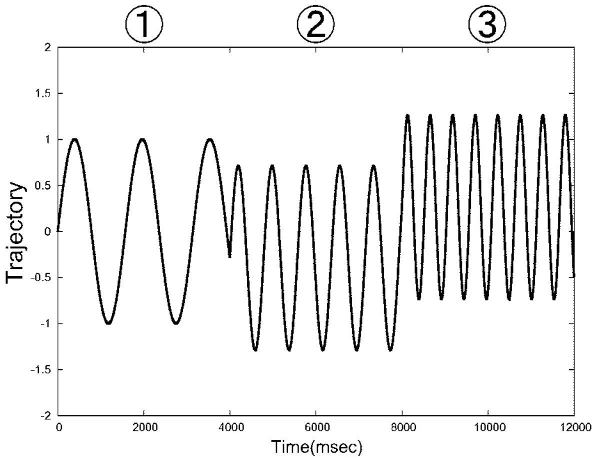

Figure 8: A desired trajectory.

	M(Kg)	B(Nm^{-1}s)	K(Nm^{-1})	NoiseVar
1	1.0	2.0	8.0	0.1
2	5.0	7.0	4.0	1.0
3	8.0	3.0	1.0	10.0

Table 2: The properties of the three environments.

section, we executed the simultaneous learning of predictors and controllers, and investigated stability of learning of p-MOSAIC. For this purpose, we prepared a complex trajectory (Fig. 8) as the desired one. In addition, the noise level for the actual state $\tilde{x}_t$ varied in each environment (Table 2).

Figure 9 shows the result obtained by p-MOSAIC where we can see that p-MOSAIC achieved appropriate parameter estimation for predictors and controllers. However, control performance (accuracy in tracking) was degraded compared with those in the former experiments (Fig 6, 7), due to the difficulty in the applied task. On the other hand, MOSAIC failed to adapt and control in most cases in this difficult setting (data not shown, because the mean square error diverged).

5 Summary

In this study, we proposed p-MOSAIC, a probabilistic model of MOSAIC, and derived learning rules based on the online EM algorithm. Our p-MOSAIC achieved an appropriate estimation of responsibility in the predictor, and an accurate control and robust learning when the controllers learn. In the adaptive control task in which MOSAIC failed, p-MOSAIC achieved stable learning. These preferable characters

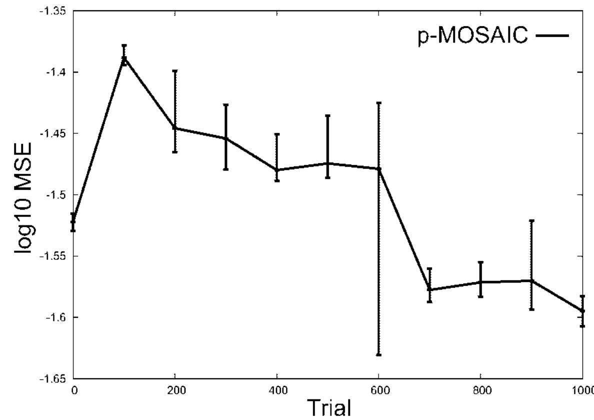

Figure 9: The logarithm of mean square error against trials.

come partly from the estimation of the inverse variance associated with the noise level of the environment, which is effective especially in noisy environments.

References

[1] C.M. Bishop. (2000). *Neural Networks for Pattern Recognition.* Oxford University Press.

[2] D.M. Wolpert and M. Kawato. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, **11**, 1317-1329.

[3] M. Sato and S. Ishii. (2000). On-line EM algorithm for the normalized Gaussian network. *Neural Computation*, **12**(2), 407-432.

[4] M. Haruno, D.M. Wolpert, M. Kawato. (2001). MOSAIC model for Sensorimotor Learning and Control. *Neural Computation*, **13**, 2201-2220.

[5] M. I. Jordan. (1998). *Learning in Graphical Models (Adaptive Computation and Machine Learning).* The MIT Press.

[6] M. Kawato. (1990). Feedback-error-learning neural network for supervised learning. In R. Eckmiller (Ed.), *Advanced neural computers* (pp. 365-372). North-Holland.

[7] R.B. Welch, B. Bridgeman, S. Anand and K.E. Browman. (1993). Alternating prism exposure causes dual adaptation and generalization to a novel displacement. *Perception and Psychophysics*, **54**(2), 195-204.