

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220500102>

MOSAIC Model for sensorimotor learning and control

Article in *Neural Computation* · October 2001

DOI: 10.1162/089976601750541778 · Source: DBLP

CITATIONS

647

READS

635

3 authors, including:



Daniel M Wolpert

University of Cambridge

230 PUBLICATIONS 25,531 CITATIONS

[SEE PROFILE](#)



Mitsuo Kawato

Advanced Telecommunications Research Institute

649 PUBLICATIONS 37,061 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



The effect of ageing on motor control [View project](#)



Computational model of cerebellum [View project](#)

MOSAIC Model for Sensorimotor Learning and Control

Masahiko Haruno

ATR Human Information Processing Research Laboratories, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

Daniel M. Wolpert

Sobell Department of Neurophysiology, Institute of Neurology, University College London, London WC1N 3BG, U.K.

Mitsuo Kawato

Dynamic Brain Project, ERATO, JST, Kyoto, Japan, and ATR Human Information Processing Research Laboratories, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

Humans demonstrate a remarkable ability to generate accurate and appropriate motor behavior under many different and often uncertain environmental conditions. We previously proposed a new modular architecture, the modular selection and identification for control (MOSAIC) model, for motor learning and control based on multiple pairs of forward (predictor) and inverse (controller) models. The architecture simultaneously learns the multiple inverse models necessary for control as well as how to select the set of inverse models appropriate for a given environment. It combines both feedforward and feedback sensorimotor information so that the controllers can be selected both prior to movement and subsequently during movement. This article extends and evaluates the MOSAIC architecture in the following respects. The learning in the architecture was implemented by both the original gradient-descent method and the expectation-maximization (EM) algorithm. Unlike gradient descent, the newly derived EM algorithm is robust to the initial starting conditions and learning parameters. Second, simulations of an object manipulation task prove that the architecture can learn to manipulate multiple objects and switch between them appropriately. Moreover, after learning, the model shows generalization to novel objects whose dynamics lie within the polyhedra of already learned dynamics. Finally, when each of the dynamics is associated with a particular object shape, the model is able to select the appropriate controller before movement execution. When presented with a novel shape-dynamic pairing, inappropriate activation of modules is observed followed by on-line correction.

1 Introduction

Given the multitude of contexts, that is, the properties of objects in the world and the prevailing environmental conditions, there are two qualitatively distinct strategies to motor control and learning. The first is to use a single controller, which would need to be highly complex to allow for all possible scenarios. If this controller were unable to encapsulate all the contexts, it would need to adapt every time the context of the movement changed before it could produce appropriate motor commands. This would produce transient and possibly large performance errors. Alternatively, a modular approach can be used in which multiple controllers coexist, with each controller suitable for one or a small set of contexts. Such a modular strategy has been introduced in the mixture of experts architecture for supervised learning (Jacobs, Jordan, Nowlan, & Hinton, 1991; Jacobs, Jordan, & Barto, 1991; Jordan & Jacobs, 1992). This architecture comprises a set of expert networks and a gating network that performs classification by combining each expert's output. These networks are trained simultaneously so that the gating network splits the input space into regions in which particular experts can specialize.

To apply such a modular strategy to motor control, two problems must be solved. First, how does the set of inverse models (controllers) learn to cover the contexts that might be experienced (the module learning problem)? Second, given a set of such inverse models how are the correct subsets selected for the current context (the module selection problem)? From human psychophysical data, we know that such a selection process must be driven by two distinct processes: feedforward selection based on sensory signals such as the perceived size of an object, and selection based on feedback of the outcome of a movement. For example, on picking up an object that appears heavy, feedforward selection may choose controllers responsible for generating a large motor impulse. However, feedback processes, based on contact with the object, can indicate that it is in fact light, thereby causing a switch to the inverse models appropriate for such an object.

Narendra and Balakrishnan (1997) and Narendra, Balakrishnan, and Ciliz (1995) have conducted pioneering work on a multiple-module system consisting of paired forward and inverse models. At any given time, the system selects only one control module, that is, an inverse model, according to the accuracy of the prediction of its paired forward model. Under this condition, they were able to prove stability of the system when the modules are used in the context of model reference control of an unknown linear time-invariant system. To guarantee this stability, Narendra's model has no learning of the inverse models and little in the forward models. In addition, no mixing of the outputs of the inverse models is allowed, thereby limiting generalization to novel contexts.

A hallmark of biological control is the ability to learn from a naive state and generalize to novel contexts. Therefore, we have sought to examine the modular architecture in a framework in which the forward and inverse mod-

els are learned *de novo*, and multiple inverse models can simultaneously contribute to control. In addition, switching between modules, as well as being driven by prediction errors as in Narendra's model, can be initiated by purely sensory cues. Therefore, our goal is to develop an empirical system that shows suitable generalization, can learn both predictors and controllers from a naive state, and can switch control based on both prediction errors during a movement and sensory cues prior to movement initiation.

In an attempt to incorporate these goals into a model of motor control and learning, Gomi and Kawato (1993) combined the feedback-error-learning (Kawato, 1990) approach and the mixture of experts architecture to learn multiple inverse models for different manipulated objects. They used both the visual shape of the manipulated objects and intrinsic signals, such as somatosensory feedback and efference copy of the motor command, as the inputs to the gating network. Using this architecture, it was possible, although quite difficult, to acquire multiple inverse models. This difficulty arose because a single gating network needed to divide up the large input space into complex regions based solely on the control error. Furthermore, Gomi and Kawato's (1993) model cannot demonstrate feedforward controller selection prior to movement execution.

Wolpert and Kawato (1998) proposed a conceptual model of sensorimotor control that addresses these problems and can potentially solve the module learning and selection problems in a computationally coherent manner. The basic idea of the model, termed the modular selection and identification for control (MOSAIC) model, is that the brain contains multiple pairs (modules) of forward (predictor) and inverse (controller) models. Within a module, the forward and inverse models are tightly coupled during both their acquisition and use. The forward models learn to divide up the contexts experienced so that for any given context, a set of forward models can predict the consequences of a given motor command. The prediction errors of each forward model are then used to gate the learning of its paired inverse models. This ensures that within a module, the inverse model learns the appropriate control for the context in which its paired forward model makes accurate predictions. The selection of the inverse models is derived from the combination of the forward model's prediction errors and the sensory contextual cues, which enables the MOSAIC model to select controllers prior to movement.

This article extends the MOSAIC model and provides the first simulations of its performance. First, the model is extended to incorporate the context transitions under the Markovian assumption (hidden Markov model, HMM). The expectation-maximization (EM) learning rule for the HMM-based MOSAIC model is derived and compared with the original proposed gradient-based method. Second, we evaluate the architecture in several respects by using the same task as Gomi and Kawato's (1993). It confirms that the MOSAIC model demonstrates more efficient learning when compared to the mixture of experts architecture. Next, we examine the performance of

the MOSAIC model to novel contexts (generalization). Finally, we discuss its performance when each context is linked to a sensory stimulus and the effects of a mismatch of the sensory stimulus with context. These simulations clearly demonstrate the usefulness of combining the forward model's feedback selection and the feedforward selection based on the sensory contextual cues.

Section 2 introduces the MOSAIC model and describes its gradient-based learning method. In section 3, the Markovian assumption of context switching is introduced, and the EM learning rule is derived. Section 4 reports several simulation results by using the multiple object manipulation task (Gomi & Kawato, 1993; Haruno, Wolpert, & Kawato, 1999b). Finally, the article concludes with the discussion in section 5.

2 MOSAIC Model

2.1 Feedforward and Feedback Selection. Figure 1 illustrates how the MOSAIC model can be used to learn and control movements when the controlled object (plant) can have distinct and changing dynamics. For example, for simulations, we will consider controlling the arm when the hand manipulates objects with distinct dynamics. Central to the MOSAIC model is the notion of dividing up experience using predictive forward models (Kawato, Furukawa, & Suzuki, 1987; Jordan & Rumelhart, 1992). We consider n undifferentiated forward models that each receives the current state, x_t , and motor command, u_t , as input. The output of the i th forward model is \hat{x}_{t+1}^i , the prediction of the next state at time t ,

$$\hat{x}_{t+1}^i = \phi(w_t^i, x_t, u_t), \quad (2.1)$$

where w_t^i are the parameters of a function approximator ϕ (e.g., neural network weights) used to model the forward dynamics. These predicted next states are compared to the actual next state to determine the likelihood that each forward model accounts for the behavior of the system. Based on the prediction errors of the forward models, the likelihood, l_t^i , of the i th forward-inverse model pair (module) generating x_t is calculated by assuming the data are generated by that forward model dynamics but contaminated with gaussian noise with standard deviation σ as

$$l_t^i = P(x_t | w_t^i, u_t, i) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-|x_t - \hat{x}_t^i|^2 / 2\sigma^2}, \quad (2.2)$$

where x_t is the true state of the system. To normalize these likelihoods across the modules, the softmax function of likelihoods in equation (2.3) could

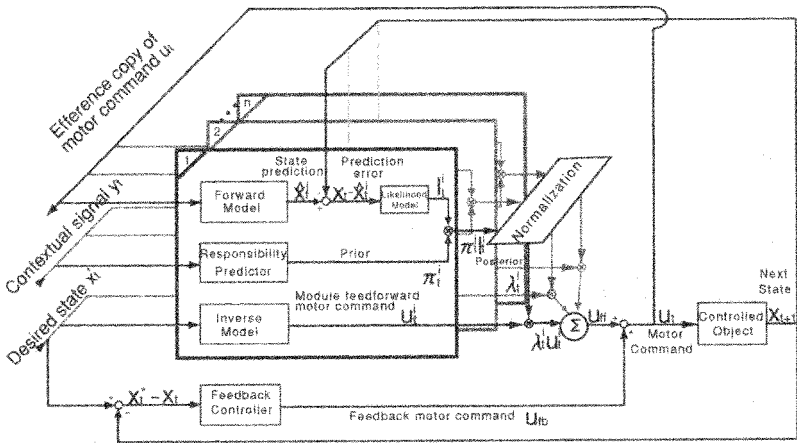


Figure 1: Schematic of the MOSAIC model with n paired modules. Each module consists of three interacting parts. The first two, the forward model and the responsibility predictor, are used to determine the responsibility of the module λ_t^i , reflecting the degree to which the module captures the current context and should therefore participate in control. The i th forward model receives a copy of the motor command u_t and makes a state prediction \hat{x}_t^i . The likelihood that a particular forward model captures the current behavior, l_t^i , is determined from its prediction error, $x_t - \hat{x}_t^i$, using a likelihood model. A responsibility predictor estimates the responsibility before movement onset using sensory contextual cues y_t and is trained to approximate the final responsibility estimate. By multiplying this estimate, the prior π_t^i , by the likelihood l_t^i , and normalizing across the modules an estimate of the module's responsibility, λ_t^i is achieved. The third part is an inverse model that learns to provide suitable control signals, u_t^i , to achieve a desired state, x_t^* , when the paired forward model provides accurate predictions. The responsibilities are used to weight the inverse model outputs to compute the final motor command u_t and control the learning within both forward and inverse models, with those models with high responsibilities receiving proportionally more of their error signal.

be used,¹

$$\frac{l_t^i}{\sum_{j=1}^n l_t^j} \tag{2.3}$$

in which a value of each module lies between 0 and 1 and sums to 1 over the modules. Those forward models that capture the current behavior, and therefore produce small prediction errors, will have large softmax values.

¹ The softmax function represents Bayes' rule in the case of the equal priors.

They could be then used to control the learning of the forward models in a competitive manner, with those models with large softmax receiving proportionally more of their error signal than modules with small softmax.

While the softmax provides information about the context during a movement, it cannot do so before a motor command has been generated and the consequences of this action evaluated. To allow the system to select controllers based on contextual information² we introduce a new component, the responsibility predictor (RP). The input to this module, y_t , contains contextual sensory information (see Figure 1), and each responsibility predictor produces a prior probability, π_t^i as in equation (2.4), that each model is responsible in a given context,

$$\pi_t^i = \eta(\delta_t^i, y_t), \quad (2.4)$$

where δ_t^i are the parameters of a function approximator η . These estimated prior probabilities can then be combined with the likelihoods obtained from the forward models by using the general framework of Bayes' rule. This produces a posterior probability as in equation (2.5),

$$\lambda_t^i = \frac{\pi_t^i l_t^i}{\sum_{j=1}^n \pi_t^j l_t^j}, \quad (2.5)$$

that each model is responsible for a given context.

We call these posterior probabilities the responsibility signals, and these form the training signal for the RP. This ensures that the priors will learn to reflect the posterior probabilities.

In summary, the estimates of the responsibilities produced by the RP can be considered as prior probabilities because they are computed before the movement execution based only on extrinsic signals and do not rely on knowing the consequences of the action. Once an action takes place, the forward models' errors can be calculated, and this can be thought of as the likelihood after the movement execution based on knowledge of the result of the movement. The final responsibility, which is the product of the prior and likelihood, normalized across the modules, represents the posterior probability. Adaptation of the RP ensures that the prior probability becomes closer to the posterior probability.

2.2 Gradient-Based Learning. This section describes gradient-descent learning of the MOSAIC model. The responsibility signal is used to gate the learning of the forward and inverse models and determine the inverse model selection. It ensures the competition between the forward models.

² We will regard all sensory information that is not an element of dynamical differential equations as contextual signal.

The following gradient-descent algorithm, equation (2.6) is similar in spirit to the “annealed competition of experts” architecture (Pawelzik, Kohlmorgen, & Müller, 1996),

$$\Delta w_t^i = \epsilon \lambda_t^i \frac{d\phi_i}{dw_t^i} (x_t - \hat{x}_t^i), \quad (2.6)$$

where ϵ is the learning rate.

For each forward model, a paired inverse model in equation (2.7) is assumed whose inputs are the desired next state x_{t+1}^* and the current state x_t . The i th inverse model produces a motor command u_t^i as output,

$$u_t^i = \psi(v_t^i, x_{t+1}^*, x_t), \quad (2.7)$$

where v_t^i are the parameters of some function approximator ψ . For physical dynamic systems, an inverse model will exist. There may not be a unique inverse, but feedback error learning, described below, will find one of the inverses that is sufficient for our model.

The total motor command in equation (2.8) is the summation of the outputs from these inverse models using the responsibilities, λ_t^i , to weight the contributions

$$u_t = \sum_{i=1}^n \lambda_t^i u_t^i = \sum_{i=1}^n \lambda_t^i \psi(v_t^i, x_{t+1}^*, x_t). \quad (2.8)$$

Therefore, each inverse model contributes in proportion to the accuracy of its paired forward model’s prediction. One of the goals of the simulations will be to test the appropriateness of this proposed mixing strategy.

Once again, the responsibilities are used to weight the learning of each inverse model as in equation (2.9). This ensures that inverse models learn only when their paired forward models make accurate predictions. Although for supervised learning the desired control command u_t^* is needed (but is generally not available), we can approximate $(u_t^* - u_t)$ with the feedback motor command signal u_{fb} (Kawato, 1990):

$$\Delta v_t^i = \epsilon \lambda_t^i \frac{d\psi_i}{dv_t^i} (u_t^* - u_t) \simeq \frac{du_t^i}{dv_t^i} \lambda_t^i u_{fb}. \quad (2.9)$$

In summary, the responsibility signals λ_t^i are used in three ways: to gate the learning of the forward models (see equation 2.6), to gate the learning of the inverse models (see equation 2.9), and to determine the contribution of the inverse models to the final motor command (see equation 2.8).

3 HMM-Based Learning

The learning rule described so far for the forward models (see equations 2.5 and 2.6) is a standard gradient-based method and thus involves the fol-

lowing drawbacks. First, the scaling parameter σ must be manually tuned. This parameter determines the extent of competition among modules and is critical for successful learning. Second, the responsibility signal λ_t^i at time t was computed only from the single observation of the sensory feedback made at time t . Without considering the dynamics of context evolution, the modules may switch too frequently due to any noise associated with the observations.

To overcome these difficulties, we can use the EM algorithm to adjust the scaling parameter σ and incorporate a probabilistic model of the context transitions. This places the MOSAIC model selection process within the framework of hidden state estimation; the computational machinery cannot directly access the true value of the responsibility, although it can observe the physical state (i.e., position, velocity, and acceleration) and the motor command. More specifically, we introduce here another way of learning and switching based on the hidden Markov model (HMM) (Huang, Ariki, & Jack, 1990). It is related to the methods proposed in the speech processing community (Poritz, 1982; Juang & Rabiner, 1985).

Let X_i be $[x_1^i, \dots, x_t^i]'$ a sequence of outputs of the i th forward model and Y_i be $[x_0^i \ u_1, \dots, x_{t-1}^i \ u_t]'$ a sequence of forward-model input of the i th module. The forward model is assumed to be a linear system with the relation $X_i = Y_i W_i$, where W_i is a column vector representing the true i th forward model parameters. By assuming gaussian noise process with a standard deviation σ_i , the likelihood $L(X|W_i, \sigma_i)$ that the i th module generates the given data is computed as follows:

$$L_i(X|W_i, \sigma_i) = \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{1}{2\sigma_i^2} (X - Y_i W_i)' (X - Y_i W_i)}.$$

To consider dynamics, HMM introduces a stationary transition probability matrix P in which a_{ij} represents a probability of switching from the i th to the j th module:

$$P = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}.$$

The gradient-based method described in the previous section implicitly assumes that $a_{ij} = 1/n$, so that all transitions are equally likely. As contexts tend to change smoothly, this assumption may lead to more frequent and noisier switching than desired. By using these notations, the total expected likelihood $L(X|\theta)$ that the n modules generate the given data is computed as in equation (3.1):

$$L(X | \theta) = \sum_{\xi} \prod_{t=0}^{T-1} a_{s_{t-1}s_t} L_{s_t}(X(t) | W_{s_t}, \sigma_{s_t}), \quad (3.1)$$

in which θ represents parameters of forward models (W_i and σ_i), and ξ and s_t are all the possible sequences of modules and a module selected at time t , respectively.

The objective here is to determine the optimal parameters P , W_i , and σ_i , which maximize the expected likelihood $L(X | \theta)$. The standard iterative technique for the problem is the Baum-Welch algorithm (Baum, Petrie, Soules, & Weiss, 1970) which is a specialized instance of the EM (Dempster, Laird, & Rubin, 1977).

The first step (E-step) is to compute the expectation in equation 3.1. To avoid the combinatorial explosion in enumerating ξ , the Baum-Welch algorithm introduces $\alpha_t(i)$ and $\beta_t(i)$ for dynamic programming. $\alpha_t(i)$ represents the forward probability $Pr(X_1, \dots, X_t, s_t = i | \theta)$ of the partial observation sequence to time t and module i , which is reached at time t given parameter θ . $\beta_t(i)$, on the other hand, is the backward probability $Pr(X_{t+1}, \dots, X_T | s_t = i, \theta)$ of the partial observation sequence from $t + 1$ to the final observation T , given module i at time t and parameter θ . The two variables can be computed recursively as follows, and the expectation is reduced to the sum of products of $\alpha_t(i)$ and $\beta_t(i)$:

$$\begin{aligned}\alpha_t(i) &= \sum_{j=1}^n \alpha_{t-1}(j) a_{ji} L_i(X(t)) \\ \beta_t(i) &= \sum_{j=1}^n \beta_{t+1}(j) a_{ij} L_j(X(t+1)) \\ L(X | \theta) &= \sum_{i=1}^n \alpha_{T-1}(i) = \sum_{i=1}^n \alpha_t(i) \beta_t(i).\end{aligned}$$

In the framework of MOSAIC model, the weighted average of inverse model outputs is necessary to compute the motor command. Equation 3.2 replaces equation 2.8 to compute the motor command:

$$u_t = \sum_{i=1}^n \alpha_{t-1}(i) u_t^i = \sum_{i=1}^n \alpha_{t-1}(i) \psi(v_t^i, x_{t+1}^*, x_t). \quad (3.2)$$

The second step (M-step) determines the optimal parameters for the current value of the expectation. We introduce two variables, $\gamma_t(i)$ and $\gamma_t(i, j)$, each of which represents the posterior probability of being in module i at time t , given the observation sequence and parameters, and a posterior probability of transition from module i to module j , conditioned on the observation sequence and parameters, respectively. Because $\gamma_t(i)$ can be regarded as an example of λ_t^i in the previous section, we can also use $\gamma_t(i)$ (and $\alpha_t(i)$ for on-line computation) instead of λ_t^i in the rest of the article:

$$\gamma_t(i) = P(s_t = i | X, \theta) = \alpha_t(i) \beta_t(i) / L(X | \theta)$$

$$\gamma_t(i, j) = P(s_t = i, s_{t+1} = j, | X, \theta) \alpha_t(i) a_{ij} L_j(X(t+1)) \beta_{t+1}(j) / L(X | \theta).$$

With these definitions, the state transition probabilities are reestimated as in equation 3.3,

$$\hat{a}_{ij} = \frac{\sum_{t=0}^{T-2} \gamma_t(i, j)}{\sum_{t=0}^{T-2} \gamma_t(i)}, \quad (3.3)$$

which is the number of expected transitions from module i to module j divided by the expected number, which stay in module i .

Since the maximization step for linear parameters is a maximum likelihood estimation weighted by $\gamma_t(i)$ (Poritz, 1982; Fraser & Dimitriadis, 1993), new estimates of W and σ (\hat{W} and $\hat{\sigma}$, respectively) are computed as in equations 3.4 and 3.5,

$$\hat{W}_i = (Y' \gamma(i) Y)^{-1} Y' X \quad (3.4)$$

$$\hat{\sigma}_i^2 = \frac{(X - Y \hat{W}_i)' \gamma(i) (X - Y \hat{W}_i)}{\sum_{t=0}^{T-1} \gamma_t(i)}, \quad (3.5)$$

which are the standard equations for weighted least-squared method (Jordan & Jacobs, 1992). In the equations, $\gamma(i)$ represents a vector of $\gamma_t(i)$. These new parameters are used to recompute the likelihood in equation 3.1. Although we discuss a batch algorithm for simplicity, on-line estimation schemes (Krishnamurthy & Moore, 1993) can be used in the same way (see Jordan & Jacobs, 1994, for details)

4 Simulation of Arm Tracking While Manipulating Objects _____

4.1 Learning and Control of Different Objects. To examine motor learning and control in the MOSAIC model, we simulated a task in which the hand had to track a given trajectory (30 sec shown in Figure 3b), while holding different unknown objects (see Figure 2). Each object had a particular mass, damping, and spring constant (M, B, K), illustrated in Figure 2. The manipulated object was periodically switched every 5 seconds among three different objects α, β , and γ in this order. The task was exactly the same as that of Gomi and Kawato (1993). We assumed the existence of a perfect inverse dynamic model of the arm for the control of reaching movements. The controller therefore needs to learn the motor commands to compensate for the dynamics of the different objects. The sampling rate was 1000 Hz, and a trial was a 30 second run. The inputs to the forward models were the motor command and the position and velocity of the hand. Each forward model outputs a prediction of the acceleration of the hand at the next time step. The inputs to the inverse model were the hand's position, velocity,

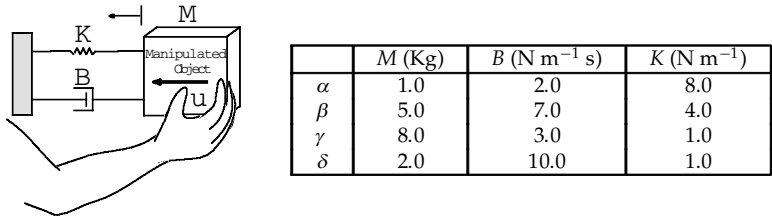


Figure 2: Schematic illustration of the simulation experiment in which the arm makes reaching movements while grasping different objects with mass M , damping B , and spring K . The object properties are shown in the table.

and acceleration of the desired state. Each inverse model output a motor command.

In the first simulation, we used the gradient-based method to train three forward-inverse model pairs (modules): the same number of modules as the number of objects. The learning rate for all modules was set to 0.01. In each module, both the forward (ϕ in equation 2.1) and inverse (ψ in equation 2.7) models were implemented as a linear neural network. The scaling parameter σ was tuned by hand over the course of the simulation. The use of linear networks allowed M , B , and K to be estimated from the forward and inverse model weights.³ Let M_j^F, B_j^F, K_j^F be the estimates from the j th forward model and M_j^I, B_j^I, K_j^I be the estimates from the j th inverse model.

Figure 3a shows an evolution of the forward model estimates of M_j^F, B_j^F, K_j^F for the three modules during learning. The three modules started from randomly selected initial conditions (open arrows) and converged over 200 trials to very good approximations⁴ of the three objects (filled arrows), as

³ The object dynamics can be described by a linear differential equation whose coefficients are defined by M , B , and K . By comparing these coefficients with weights of the linear network, we can calculate the estimated value of M , B , and K from both forward and inverse weights.

⁴ The weights were assumed to be converged if the predicted acceleration error per point became less than 0.001.

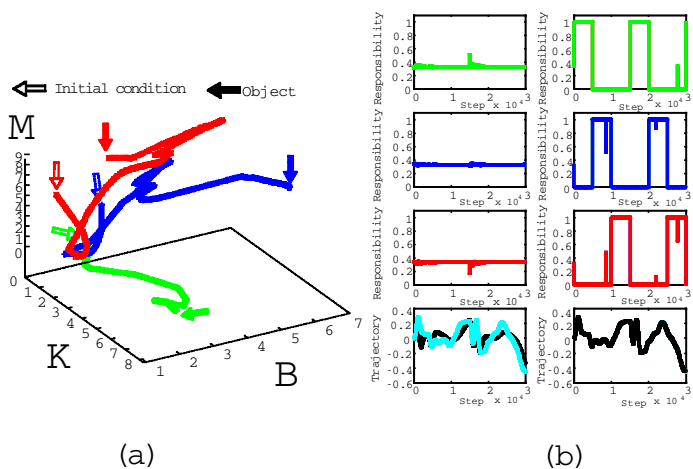


Figure 3: (a) Learning of three forward models is represented in terms of the parameters of the three corresponding objects. (b) Responsibility signals from the three modules (top 3) and tracking performance (bottom: gray is the desired trajectory) before (left) and after (right) learning.

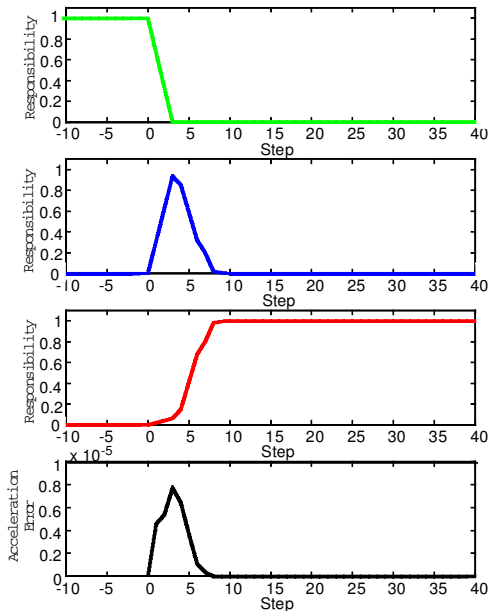


Figure 5: Responsibility predictions based on contextual information of two-dimensional object shapes (top three traces) and corresponding acceleration error of control induced by an unusual shape-dynamic pairing (bottom trace).

Table 1: Learned Object Characteristics.

Module	M_j^F	B_j^F	K_j^F	M_j^I	B_j^I	K_j^I
1	1.0020	2.0080	8.0000	1.0711	2.0080	8.0000
2	5.0071	7.0040	4.0000	5.0102	6.9554	4.0089
3	8.0029	3.0010	0.9999	7.8675	3.0467	0.9527

shown in Table 1. Each of the three modules converged to α , β , and γ objects, respectively. It is interesting to note that the estimates of the forward models are superior to those of the inverse models. This is because the inverse model learning depends on how modules are switched by the forward models.

Figure 3b shows the performance of the model before (left) and after (right) learning. The top three panels show, from top to bottom, the responsibility signals of the first, second, and third modules, and the bottom panel shows the hand's actual and desired trajectories. At the start of learning, the three modules were equally poor and thus generated almost equal responsibilities (one-third) and were equally involved in control. As a result, the overall control performance was poor, with large trajectory errors. However, at the end of learning, the three modules switched quite well but did show occasional inappropriate spikes even though the learning parameter was carefully chosen. If we compare these results with Figure 7 of Gomi and Kawato (1993) for the same task, the superiority of MOSAIC model compared to the gating expert architecture is apparent. Note that the number of free parameters (synaptic weights) is smaller in the current architecture than the other. The difference in performance comes from two features of the basic architecture. First, in the gating architecture, a single gating network tries to divide the space while many forward models split the space in the MOSAIC model. Second, in the gating architecture, only a single control error is used to divide the space, but multiple prediction errors are simultaneously used in the MOSAIC model.

4.2 Feedforward Selection. This section describes results of a simple simulation of feedforward selection using responsibility predictors. Each dynamic object was associated with a visual cue, A, B, or C as shown in Figure 4. The visual cue was a two-dimensional (2-D) shape represented as a 3×3 binary matrix, which was randomly placed at one of four possible locations on a 4×4 retinal matrix. The retinal matrix was used as the contextual input to the RP (three-layer sigmoid feedforward network with 16 inputs, 8 hidden units, and 3 output units). During the course of learning, the combination of manipulated objects and visual cues was fixed as A- α , B- β , and C- γ . After 200 iterations of the trajectory, the combination A- γ was presented for the first time. Figure 5 plots the responsibility signals of the three modules (top three traces) and corresponding acceleration error

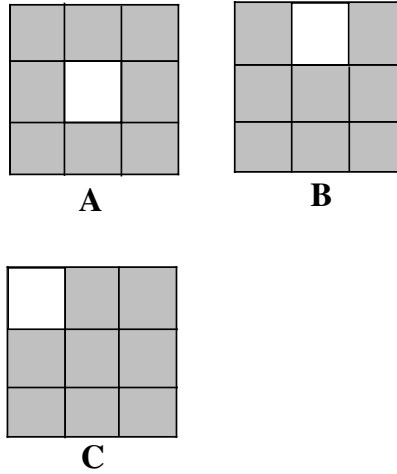


Figure 4: Visual representations of the objects used as input to the responsibility predictors.

of the control induced by the conflicting visual-dynamic pairing (bottom trace). Until the onset of movement (time 0), *A* was always associated with light mass of α , and *C* was always associated with the heavy mass γ . Prior to movement when *A* was associated with γ , the α module was switched on by the visual contextual information, but soon after the movement was initiated, the likelihood signal from the forward model's prediction dominated, and the γ module was properly selected. This demonstrates how the interplay of feedforward and feedback selection processes can lead to on-line reselection of appropriate control modules.

4.3 Feedback Selection by HMM. We will show here that the MOSAIC model with the EM algorithm was able to learn both the forward and inverse models and the transition matrix. In particular, when the true transitions were made cyclic or asymmetric, the estimated transition matrix closely captured the true transitions. For example, when we tested a hidden state dynamics generated by the transition matrix P , the EM algorithm converged to the following transition matrix \hat{P} , which was very close to the true P :

$$P = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.99 & 0.01 \\ 0.01 & 0 & 0.99 \end{pmatrix}$$

$$\hat{P} = \begin{pmatrix} 0.9816 & 0.0177 & 0.0007 \\ 0.0021 & 0.9882 & 0.0097 \\ 0.0079 & 0.0000 & 0.9920 \end{pmatrix}.$$

Table 2: Acceleration Error Rates: HMM-MOSAIC/Gradient-MOSAIC with Two Types of Initial Conditions.

Initial Conditions S.D.	Switching Period (Steps)				
	5	25	50	100	200
-					
20%	0.998	0.887	1.031	0.636	0.443
40%	0.855	0.676	1.147	0.598	0.322

Now, we compare the gradient-based (see equation 2.6) and EM (see equation 3.4) learning methods on the same condition as Figure 2. We tested five switching periods of 5, 25, 50, 100, and 200 steps for the total trial length of 600 steps. Separate simulations were run in which the initial weights of the inverse and forward models were generated by random processes with two different standard deviations.⁵ The scaling parameter σ in the gradient-based method was fixed through the simulation, whereas the EM method automatically adjusted this parameter. Fifty repetitions of 20 trials were performed for each pair of switching period and initial weight variance.

Table 2 shows the ratio of the final acceleration errors of the two learning methods: the error of EM algorithm divided by the error of gradient-based method. The EM version of the MOSAIC model (HMM-MOSAIC) provides more robust estimation in two senses. First, HMM-MOSAIC achieved a precise estimation for a broad range of switching periods. In particular, it significantly outperforms gradient-MOSAIC for infrequent switching. This is probably because gradient-MOSAIC does not take the switching dynamics into account and switches too frequently. Figure 6 exemplifies typical changes in responsibility and actual acceleration for the two learning methods. It confirms the more stable switching and better performance of HMM-MOSAIC. The second superiority of HMM-MOSAIC is the better performance when the learning starts from worse (larger variance) initial values. These two results, with the additional benefit that the scaling parameter, σ , need not be hand-tuned, confirms that HMM-MOSAIC is more suitable for learning and detecting the switching dynamics if the true switching is infrequent (i.e., every more than 100 steps).

4.4 Generalization to a Novel Object. A natural question regarding the MOSAIC model is how many modules need to be used. In other words, what happens if the number of objects exceeds the number of modules or an already trained MOSAIC model is presented with an unfamiliar object? To examine this, HMM-MOSAIC, trained with four objects α , β , γ , and

⁵ The standard deviation of the initial conditions were 20% and 40%, respectively. In both cases, the mean was equal to the true parameters.

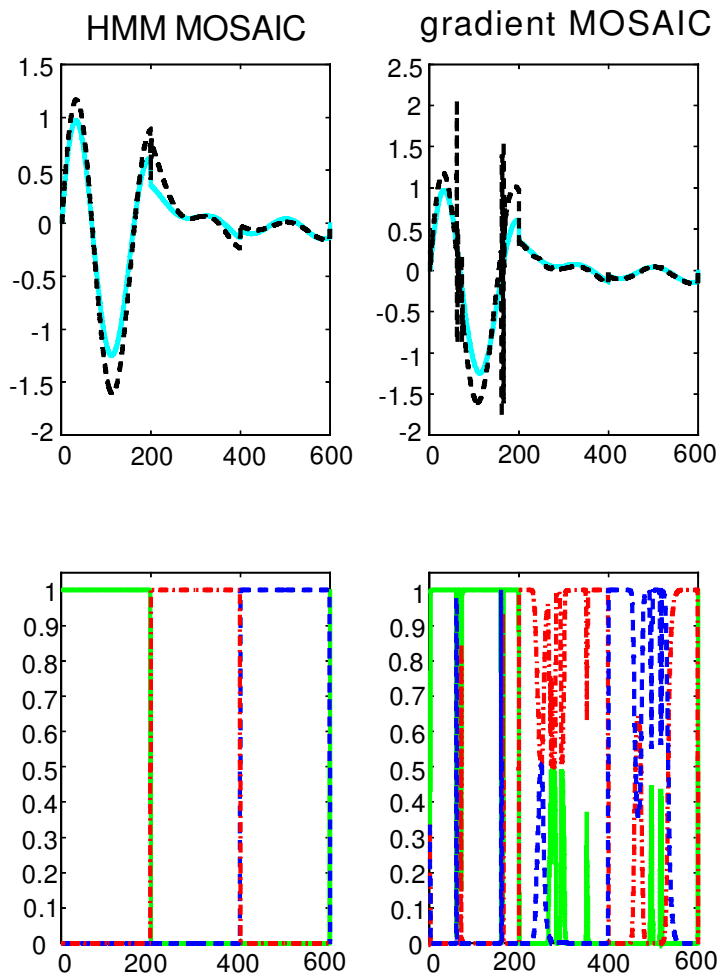


Figure 6: Actual accelerations and responsibilities of HMM-MOSAIC and gradient-MOSAIC. (Top panels) Actual (black) and desired (blue) accelerations. (Bottom Panels) Responsibilities of the three modules.

δ in Figure 2, was presented with a novel object η (its (M, B, K) is $(2.61, 5.76, 4.05)$). Because the object dynamics can be represented in a three-dimensional parameter space and the four modules already acquired define

four vertices of a tetrahedron within the 3D space, arbitrary object dynamics contained within the tetrahedron can be decomposed into a weighted average of the existing four forward modules (internal division point of the four vertices). The theoretically calculated linear weights of η were (0.15,0.20,0.35,0.30). Interestingly, each module's responsibility signal averaged over trajectory was (0.14,0.25,0.33,0.28), and the mean squared Euclidean deviation from the true value was 0.16. This combination produced the error that was comparable to the performance with the already learned objects. When tested on 50 different objects within the tetrahedron,⁶ the average Euclidean distance between the true internal division vector and estimated responsibility vector was 0.17. Therefore, although the responsibility was computed in the space of acceleration and had no direct relation to the space of (M, B, K) , the two vectors had very similar values. This demonstrates the flexible adaptation of the MOSAIC model to unknown objects, which originates from its probabilistic soft-switching mechanism. This is in sharp contrast to the hard switching of Narendra and Balakrishnan (1997) in which only one module can be selected at a time. Extrapolation outside the tetrahedron is likely to produce poorer performance, and it is an open research question as to the conditions under which extrapolation could be achieved and how the existing modules relearn in such a situation.

5 Discussion

We have proposed a novel modular architecture, the MOSAIC model for motor learning and control. We have shown through simulation that this model can learn the controllers necessary for different contexts, such as object dynamics, and switch between them appropriately based on both sensory cues such as vision and feedback. The architecture also has the ability to generalize to novel dynamics by blending the outputs from its learned controllers. The architecture can also correct on-line for inappropriate selection based on erroneous visual cues.

We introduced the Markovian assumption primarily for computational efficiency. However, this can also be viewed as a model of the way context is estimated in human motor control. Vetter and Wolpert (2000b) have examined context estimation in humans and shown that the central nervous system (CNS) estimates the current context using information from both prior knowledge of how the context might evolve over time and from the comparison of predicted and actual sensory feedback. In one experiment, subjects learned to point to a target under two different visuomotor contexts, for example, rotated and veridical visual feedback. The contexts were

⁶ Since responsibility signals always take a positive value, we will focus on the inside of the tetrahedron.

chosen so that at the start of the movement, subjects could not tell which context they were acting in. After learning, the majority of the movement was made without visual feedback. However, during these movements, brief instances of visual feedback were provided that corresponded to one of the two contexts. The relationship between the feedback provided and the subjects' estimate of the current movement context was examined by measuring where they pointed relative to where they should point for each of the different contexts. The results showed that in the presence of uncertainty about the current context, subjects produced an estimate of the current context intermediate to the two learned contexts rather than simply choose the more likely context. There were also systematic relationships between the timing and nature of the feedback provided and the final context estimate. A two-state HMM provided a very good fit to these data. This model incorporates an estimate of the context transition matrix and sensory feedback process, visual location corrupted by noise, and optimally estimates the current context. Vetter and Wolpert also studied whether the CNS could model the prior probability of context transitions. This was examined in cases where the context changed either discretely between movement (Vetter & Wolpert, 2000b) or continuously during a movement (Vetter & Wolpert, 2000a). In both cases, when feedback was withheld, subjects' estimate of context showed behavior that demonstrated that the CNS was using knowledge about context transitions to estimate the context in the absence of feedback. Taken together, these results support that the CNS uses a model of context transition and compares predicted and actual sensory feedback to estimate the current context. Although their HMM model does not contain inverse models and responsibility predictors, it is very close in spirit to our forward models. The HMM was also tested as a model of human context estimation when either the context transition or sensory feedback noise was removed. The result is compatible with our simulation in that neither could explain the experimental data.

One may ask how the MOSAIC model works when the context evolution has a hierarchical structure of depth greater than one. A natural extension of the MOSAIC model is to a hierarchical structure that consists of several layers, each comprising a MOSAIC model. At each layer, the higher-level MOSAIC model interacts with its subordinate MOSAIC models through access to the lower-level MOSAIC model's responsibility signals, representing which module should be selected in the lower level given the current behavioral situation. The higher level generates descending commands, which represent the prior probability of the responsibility signal for the lower-level modules, and thus prioritizes which lower-level modules should be selected. This architecture could provide a general framework that is capable of sequential motor learning, chunking of movement patterns, and autonomous symbol formation (Haruno, Wolpert, & Kawato, 1999a).

6 Conclusion

We have described an extension and evaluation of the MOSAIC model. Learning was augmented by the EM algorithm for HMM. Unlike gradient descent, it was found to be robust to the initial starting conditions and learning parameters. Second, simulations of an object manipulation task demonstrate that our model can learn to manipulate multiple objects and to switch among them appropriately. Moreover, after learning, the model showed generalization to novel objects whose dynamics lay within the tetrahedron of already learned dynamics. Finally, when each of the dynamics was associated with a particular object shape, the model was able to select the appropriate controller before movement execution. When presented with a novel shape-dynamic pairing, inappropriate activation of modules was observed, followed by on-line correction. In conclusion, the proposed MOSAIC model for motor learning and control, like the human motor system, can learn multiple tasks and shows generalization to new tasks and an ability to switch between tasks appropriately.

Acknowledgments

We thank Zoubin Ghahramani for helpful discussions on the Bayesian formulation of this model. This work was partially supported by Special Coordination Funds for promoting Science and Technology at the Science and Technology Agency of the Japanese government, the Human Frontiers Science Program, the Wellcome Trust, and the Medical Research Council.

References

- Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic function of Markov chains. *Ann. Math. Stat.*, 41, 164–171.
- Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society B*, 39, 1–38.
- Fraser, A., & Dimitriadis, A. (1993). Forecasting probability densities by using hidden Markov models with mixed states. In A. Wiegand, & N. Gershenfeld (Eds.), *Time series prediction: Forecasting the future and understanding the past* (pp. 265–282). Reading, MA: Addison-Wesley.
- Gomi, H., & Kawato, M. (1993). Recognition of manipulated objects by motor learning with modular architecture networks. *Neural Networks*, 6, 485–497.
- Haruno, M., Wolpert, D., & Kawato, M. (1999a). Emergence of sequence-specific neural activities with hierarchical multiple paired forward and inverse models. *Neuroscience Meeting Abstract*, p. 1910 (760.2).
- Haruno, M., Wolpert, D., & Kawato, M. (1999b). Multiple paired forward-inverse models for human motor learning and control. In M. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in neural information processing systems*, 11 (pp. 31–37). Cambridge, MA: MIT Press.

- Huang, X., Ariki, Y., & Jack, M. (1990). *Hidden Markov models for speech recognition*. Edinburgh: Edinburgh University Press.
- Jacobs, R., Jordan, M., & Barto, A. (1991). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*, 15, 219–250.
- Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive mixture of local experts. *Neural Computation*, 3, 79–87.
- Jordan, M., & Jacobs, R. (1992). Hierarchies of adaptive experts. In J. Moody, S. Hanson, & R. Lippmann (Eds.), *Advances in neural information processing systems*, 4 (pp. 985–993). San Mateo, CA: Morgan Kaufmann.
- Jordan, M., & Jacobs, R. (1994). Hierarchical mixture of experts and the EM algorithm. *Neural Computation*, 6, 181–214.
- Jordan, M., & Rumelhart, D. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16, 307–354.
- Juang, B., & Rabiner, L. (1985). Mixture autoregressive hidden Markov models for speech signals. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-33(6), 1404–1413.
- Kawato, M. (1990). Feedback-error-learning neural network for supervised learning. In R. Eckmiller (Ed.), *Advanced neural computers* (pp. 365–372). Amsterdam: North-Holland.
- Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural network model for the control and learning of voluntary movements. *Biol. Cybern.*, 56, 1–17.
- Krishnamurthy, V., & Moore, J. (1993). On-line estimation of hidden Markov model parameters based on the Kullback-Leibler Information Measure. *IEEE Transactions on Signal Processing*, 41(8), 2557–2573.
- Narendra, K., & Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transaction on Automatic Control*, 42(2), 171–187.
- Narendra, K., Balakrishnan, J., & Ciliz, M. (1995). Adaptation and learning using multiple models, switching and tuning. *IEEE Contr. Syst. Mag.*, 37–51.
- Pawelzik, K., Kohlmorgen, J., & Müller, K. (1996). Annealed competition of experts for a segmentation and classification of switching dynamics. *Neural Computation*, 8, 340–356.
- Poritz, A. (1982). Linear predictive hidden Markov models and the speech signal. In *Proc. ICASSP 82* (pp. 1291–1294).
- Vetter, P., & Wolpert, D. (2000a). The CNS updates its context estimate in the absence of feedback. *NeuroReport*, 11, 3783–3786.
- Vetter, P., & Wolpert, D. (2000b). Context estimation for sensory motor control. *Journal of Neurophysiology*, 84, 1026–1034.
- Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11, 1317–1329.

Received August 23, 1999; accepted November 22, 2000.