Electronics and Communications in Japan, Part 3, Vol. 89, No. 9, 2006
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J85-D-II, No. 1, January 2002, pp. 90–100

# Symbolization and Imitation Learning of Motion Sequence Using Competitive Modules

Kazuyuki Samejima,[1] Ken'ichi Katagiri,[2] Kenji Doya,[2,3] and Mituo Kawato[2,3]

[1]Japan Science and Technology Corp., Kyoto, 619-0288 Japan

[2]Nara Institute of Science and Technology, Ikoma, 630-0101 Japan

[3]Information Science Division, ATR-International, Kyoto, 619-0288 Japan

## SUMMARY

In this research the authors evaluate a new method for control using several prediction models and recognition of movement series. In MOSAIC (MOdule Selection And Identification for Control), which uses a prediction model with several modules as proposed by Wolpert and Kawato (1998), a module that pairs a prediction model which predicts the future state to be controlled and a controller are switched and assembled based on the size of the prediction error in the prediction model. The authors propose a method using MOSAIC to divide continuous time patterns for human or robot movement into their constituent parts as several series of movement elements. Moreover, the authors evaluate a method to recognize movement patterns of another person using one's own module and imitation learning based on this method. From the results of simulations of acrobot control, the authors show that symbolization of movement patterns and imitation learning based on that are possible. © 2006 Wiley Periodicals, Inc. Electron Comm Jpn Pt 3, 89(9): 42–53, 2006; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ecjc.20267

Contract grant sponsor: "Research on Human and Primate Communications" through the 1998 Science and Technology Advancement Funding in the Science and Technology Agency.

**Key words:** MOSAIC; symbolism; imitation learning; acrobot.

## 1. Introduction

In the real world there are several ways of achieving a given goal (efficient ones, inefficient ones, and creative ones), and not necessarily just one. There are multiple patterns for achieving an upstream climb and other movement. Different movement patterns are thought to exist because there are several movement units, there are several transition patterns for symbol series representing the movement units (action elements), and there are different transition patterns. In human learning, the goal is first achieved by repeated trial and error. However, movement patterns that vary with individuals are created as a result of failures along the way, or low levels of efficiency even when the goal is achieved.

On the other hand, if movement series performed by others (teachers) are observed, and they are used as prior information for one's own movement learning, then more efficient movement patterns can sometimes be acquired. In other words, control of learning in a nonlinear system can involve being trapped at a locally optimal solution, but by imitating successful examples, a more efficient movement series can be acquired.

However, simple imitation by using the movement trajectory of another is not straightforward. In general, movement commands such as joint motion cannot be directly observed, and even if they could be, the same movement trajectory cannot necessarily be generated based on the same movement commands because of differences in body parameters. In this type of learning process, first there must be an awareness that the movement pattern being observed (teacher) is substituted into one's own movements, and then a movement pattern matching one's own body must be generated using the results of this awareness.

The authors have previously proposed MOSAIC (MOdule Selection And Identification for Control), a control method that uses several models [1–3]. When a target trajectory is given in the MPFIM (Multiple Paired Forward-Inverse Model), the learning edition and teacher for MOSAIC, by using modules for the "reverse model" which generates the control output and the "forward model" which predicts state changes based on the control output and the current state, re-adaptation and de-adaptation for a nonconstant environment can be performed rapidly [2, 4].

In the Multiple Model-based Reinforcement Learning (MMRL), the reinforcement learning version of MOSAIC, a reward is given instead of a target trajectory. Movement can be realized by training an evaluation function, the expected value for the reward in the future [5, 6]. MMRL can train and create an optimal controller efficiently by creating a pairing using a reinforcement learning controller and a prediction model. Moreover, because complex environmental components are separated in terms of time and space by combining simple prediction models, an effective number of modules can be determined automatically based on the complexity of the environment [3].

Each module of MOSAIC is taken to represent several different operating elements. In concrete terms, by assigning a single symbol to each MOSAIC module, not only can one's own continuous movement be divided into its constituent parts, but the movement of another can be represented as symbols when replacing it with one's own.

In this paper, using MOSAIC the authors propose imitation learning in which movement patterns are acquired more efficiently by (1) a series representation of symbols for the movement pattern, (2) symbol recognition in which the movement pattern of another is recognized as a series of movement units using one's own predictor and controller, and (3) making use of the recognized movement series. The authors demonstrate that this is possible using simulations for control goals in an acrobot.

This paper is structured as follows. In Section 2, an overview of MOSAIC is given, and evaluation of the symbolization of movement patterns using MOSAIC is performed. Next, the symbol series is estimated based on the observed movement patterns, and a method for using the results for imitation is proposed. In Section 3, the simulation results for a swing-up task for the acrobot are given. Section 4 discusses MOSAIC as a model for communication in the brain, and Section 5 provides a summary of the paper.

## 2. Symbolization and Imitation Using MOSAIC

### 2.1. An overview of MOSAIC

In MOdule Selection And Identification for Control, MOSAIC [1–3], the pair consisting of a prediction model which predicts what is to be controlled and the controller which controls the trajectory are treated as one module, and several such modules are used. First, based on the prediction error of each prediction module, the "responsibility signal" which represents the adaptability of each module for the current environmental state or context is calculated. Based on the responsibility signal, the control output, prediction model, and the contribution ratio for learning in the controller are determined. Moreover, when the responsibility signal can be predicted based on some form of context information, module selection is performed using the predicted value for the responsibility signal as an a priori probability.

As can be seen in Fig. 1, MOSAIC consists of $n$ prediction modules which create predictions for the dynamic characteristics to be controlled under given conditions or using approximations of operating points and $n$ controllers for each. At each point in time, how to select or combine the output of each module, and with what probability to perform learning in each module is determined by the "responsibility signal" given by the soft-max function for the magnitude of the prediction error in each prediction model.

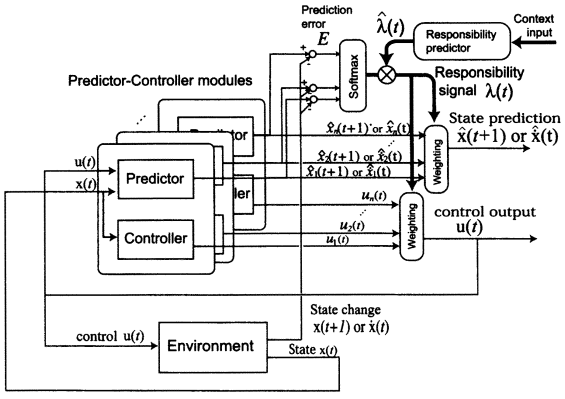


Fig. 1. MOSAIC: Module selection and identification for the control module.

An overview of the operation of the system is given below.

(1) Responsibility signal: The state changes to be controlled are predicted using the *n* prediction models, and then based on the prediction squares error $E_i(t)$, the responsibility signal $\lambda_i$ for each module is given by

$$\lambda_i(t) = \frac{e^{-\frac{E_i(t)}{2\sigma^2}}}{\sum_{j=1}^{n} e^{-\frac{E_j(t)}{2\sigma^2}}} \quad (1)$$

Here, $\sigma$ is a parameter which indicates the size of the shared range for each module. $E_i(t)$ is given by

$$E_i(t) = ||\hat{\mathbf{x}}_i(t) - \mathbf{x}(t)||^2 \quad (2)$$

which is the simplest instantaneous prediction error. Here, $\hat{\mathbf{x}}_i$ represents the output value for the *i*-th prediction model, and $\mathbf{x}(t)$ represents the real state. Moreover, for $E_i(t)$, module selection can be performed more stably by using a brief average value for the prediction error to be described later.

(2) Determining the action output: Using a weight proportional to the responsibility signal $\lambda_i(t)$, the output $\mathbf{u}_i(t) = \mu_i(\mathbf{x}(t))$ for each controller is added, and the action output to be controlled

$$\mathbf{u}(t) = \sum_{i=1}^{n} \lambda_i(t)\mathbf{u}_i(t) \quad (3)$$

results.

(3) Learning in each module: Using reinforcement proportional to the responsibility signal $\lambda_i(t)$, learning for the state prediction model $f_i(\mathbf{x}, \mathbf{u})$ and learning for the controller are performed.

As a result, optimization is performed for different operating points and different operating conditions to be controlled in a form in which consistency is maintained in switching and combining modules and in the prediction model and controller in each module.

Moreover, when prior information is given for module selection, the responsibility signal is given by

$$\lambda_i(t) = \frac{\hat{\lambda}_i(t)e^{-\frac{E_i(t)}{2\sigma^2}}}{\sum_{j=1}^{n} \hat{\lambda}_j(t)e^{-\frac{E_j(t)}{2\sigma^2}}} \quad (i = 1, 2, \ldots, n) \quad (4)$$

instead of Eq. (1) using the prediction value $\hat{\lambda}_i(t)$ for the responsibility signal based on the prior information and the prediction squares error $E_i(t)$ for the prediction model.

### 2.2. Symbolization and imitation

There is a close relationship between the operating elements due to the mutual interactions between the module structure in MOSAIC and the environment. For instance, in an example using MMRL, the reinforcement learning version of MOSAIC, for learning control of a nonlinear system, each module approximates linear dynamics at stable or unstable equilibrium points for the system, and a control (reinforcement learning controller) which varies the stability based on the objective is formed [3]. In other words, the phrasing of the movement based on the environment is formed by the responsibility signal.

Given this, the time series for the responsibility signal $\boldsymbol{\lambda}$ which determines the burden for each module can be thought of as providing a symbol representation which abstracts the movement pattern. Moreover, if the movement of another being observed can be recognized as a transition pattern for the responsibility signal $\boldsymbol{\lambda}$, then by creating a movement pattern that is in line with the recognized transition series, a movement pattern distinct from what one is performing can be imitated.

The symbolization of the movement pattern used here with MOSAIC and the method for using it for imitation learning are shown in a schematic diagram (Fig. 2). Below, (1) a method for estimating the responsibility signal $\lambda$ based on the movement pattern $\mathbf{x}^{obs}(t)$ being observed, and (2) a method for using the series $\lambda$ for imitation learning are given.

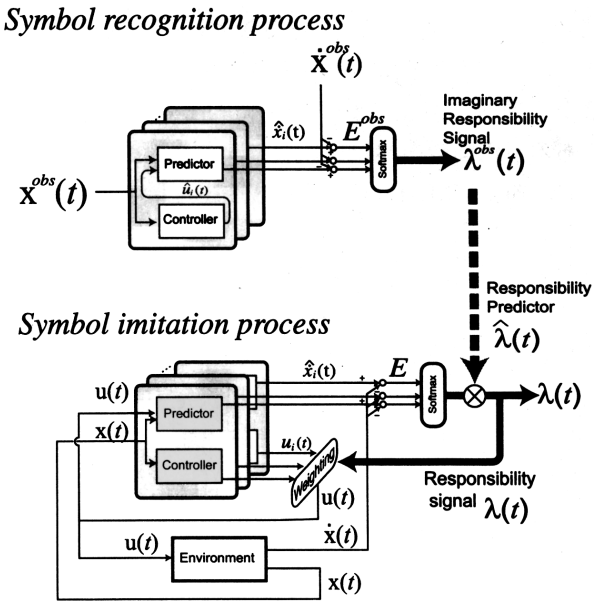


Fig. 2. Schematic diagrams of symbol recognition process (upper diagram) and symbol imitation process (lower diagram).

#### 2.2.1. Symbol recognition: Estimating the responsibility signal series based on observing movement patterns in another

Let the state being observed at the time $t$ be $\mathbf{x}^{obs}(t)$. If this is input to each controller, then based on

$$\mathbf{u}_i(t) = \mu_i(\mathbf{x}^{obs}(t)) \tag{5}$$

the output $\mathbf{u}_i(t)$ for each controller is determined (refer to Fig. 2). $\mu_i$ represents the control rules for the modules. Moreover, if the control output $\mathbf{u}_i(t)$ and the observed $\mathbf{x}^{obs}(t)$ are input into the paired prediction module $f_i$, then based on

$$\dot{\mathbf{x}}_i^{obs}(t) = f_i(\mathbf{x}^{obs}(t), \mathbf{u}_i(t)) \tag{6}$$

the changes in the state when the $i$-th module is selected can be predicted. Based on the brief average for the error between this predicted value and the state change $\dot{\mathbf{x}}^{obs}(t)$ obtained from actual observations, that is,

$$\epsilon \dot{E}_i^{obs}(t) = -E_i^{obs}(t) + ||\dot{\mathbf{x}}_i^{obs} - \dot{\mathbf{x}}^{obs}||^2 \tag{7}$$

the soft-max function is used, and based on

$$\lambda_i^{obs}(t) = \frac{e^{-\frac{E_i^{obs}}{2\sigma^2_{obs}}}}{\sum_{j=1}^{n} e^{-\frac{E_j^{obs}}{2\sigma^2_{obs}}}} \tag{8}$$

the responsibility signal for $\mathbf{x}^{obs}(t)$ can be estimated. Here, $0 < \varepsilon$ is a constant updated for the responsibility signal predicted value.

#### 2.2.2. Imitation learning using the series for the responsibility signal λ

As described above, the responsibility signal $\lambda_i^{obs}(t)$ estimated based on the observed movement pattern $\mathbf{x}^{obs}(t)$ can be taken to be the responsibility signal predicted value $\hat{\lambda}_i(t)$ for the real movement, and is used as transcendental information for the MOSAIC module selection using Eq. (4).

## 3. Simulations

### 3.1. Acrobot swing-up task

An acrobot is a robot with two links and two joints, as can be seen in Fig. 3. An actuator exists only at the second joint in the waist; there is no actuator in the first joint at the hand. The acrobot swing-up task is a very difficult task in which a nonlinear objective is moved from a stable, dangling state to an unstable standing state and then held there [7–9].

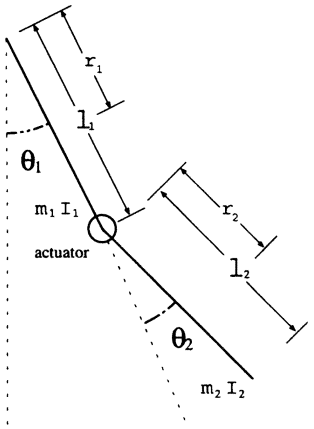


Fig. 3. An acrobot: two-link robot with an actuator only at the second joint.

The physical parameters in this experiment are listed in Table 1, and the equations of motion are given in Appendix 1.

### 3.2. MOSAIC using a linear state prediction model and quadratic form reward model

The linear model is characterized by learning speed and highly generalized capacity. Thus, in MOSAIC, a linear model was used for the prediction model and the controller. Here, the linear model for control is

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{9}$$

and the reward function is

$$r(\mathbf{x}(t), \mathbf{u}(t)) = -\frac{1}{2}\mathbf{x}'(t)Q\mathbf{x}(t) - \frac{1}{2}\mathbf{u}'(t)R\mathbf{u}(t) \tag{10}$$

both of which are assumed to be in quadratic form. The evaluation function $V(\mathbf{x})$, the time attenuation integration of the reward function, is based on the matrix $P$, a solution to the Riccati equation

$$0 = -PA - A'P + PBR^{-1}B'P - Q \tag{11}$$

Table 1. Parameters of the acrobot

| | link1 | link2 |
|---|---|---|
| Length: $l$ | 1 m | 1 m |
| Distance to the center of gravity: $r$ | 0.5 m | 0.5 m |
| Mass: $m$ | 1 kg | 1 kg |
| Inertial moment: $I$ | 1 kg · m$^2$ | 1 kg · m$^2$ |

and is given by

$$V(\mathbf{x}(t)) = -\frac{1}{2}\mathbf{x}'(t)P\mathbf{x}(t) \tag{12}$$

here [10]. Based on this evaluation function [Eq. (12)] and the linear model [Eq. (9)] for control, the optimal feedback control rule which maximizes the evaluation is given by

$$\mathbf{u}(t) = -R^{-1}B'P\mathbf{x}(t) = -K\mathbf{x}(t) \tag{13}$$

When using this type of linear quadratic controller (LQC) in each controller, the local linear prediction module

$$\hat{\dot{\mathbf{x}}}_i(t) = A_i(\mathbf{x}(t) - \mathbf{x}_i) + B_i\mathbf{u}(t) \tag{14}$$

and a local quadratic reward model

$$\hat{r}_i(\mathbf{x}(t), \mathbf{u}(t)) = -\frac{1}{2}(\mathbf{x}(t) - \mathbf{x}_i)'Q_i(\mathbf{x}(t) - \mathbf{x}_i) - \frac{1}{2}\mathbf{u}(t)'R_i\mathbf{u}(t), \tag{15}$$

are prepared for each module. Here, $\mathbf{x}_i$ is the center point for local approximation. The Riccati equation

$$0 = -P_iA_i - A_i'P_i + P_iB_iR_i^{-1}B_i'P_i - Q_i \tag{16}$$

is solved for these models, and the feedback gain matrix for each module

$$K_i = R_i^{-1}B_i'P_i \tag{17}$$

can be found. The control output is assumed to be a weighting of the output for each LQC using the responsibility signal $\lambda_i(t)$, that is,

$$\mathbf{u}(t) = -\sum_{i=1}^{n}\lambda_i(t)K_i(\mathbf{x}(t) - \mathbf{x}_i) \tag{18}$$

### 3.2.1. Experimental methods

It has been shown that the prediction model can acquire through learning a linear learning model that approximates dynamics when an acrobot learns the swing-up [3]. Thus, in this research the authors prepare the coefficient matrices $A_i$ and $B_i$ for the linear prediction model near equilibrium points analytically and not through learning in order to emphasize generation and recognition of movement patterns.

Figure 4 represents the posture at four acrobot equilibrium points: $\mathbf{x}_1 = (0, 0, 0, 0)'$, $\mathbf{x}_2 = (0, \pi, 0, 0)'$, $\mathbf{x}_3 = (\pi, \pi, 0, 0)'$, and $\mathbf{x}_4 = (\pi, 0, 0, 0)'$. When there is no control input at the four equilibrium points, the prediction model is provided with the linear form in Eq. (14) of Eqs. (A.1) and (A.2).

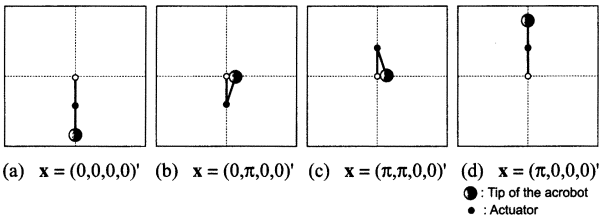


Fig. 4. Postures of acrobot at four equilibrium points.

The reward $r(t)$ is taken to be

$$r(t) = -\frac{1}{2}(m_1r_1\cos\theta_1 + m_2l_1\cos\theta_1 + m_2r_2\cos(\theta_1 + \theta_2)) - \frac{1}{2}Ru^2 \tag{19}$$

that is the sum of the link position energy and the torque squared, where $R$ is varied. This equation can also be represented in the form

$$r(\mathbf{x}(t), \mathbf{u}(t)) = -\frac{1}{2}(\mathbf{x}(t) - \mathbf{x}_i)'Q(\mathbf{x}(t) - \mathbf{x}_i) - \frac{1}{2}\mathbf{u}'(t)R\mathbf{u}(t) \tag{20}$$

when quadratic approximation is performed at the four equilibrium points described above, representing a quadratic form for the state $\mathbf{x}$ and the control input $\mathbf{u}$. A coefficient matrix $Q_i$ for the reward near the four equilibrium points $\mathbf{x}_i$ was also found analytically as was done for $A_i$ and $B_i$.

In this fashion, based on the coefficients $A_i$ and $B_i$ [refer to Eq. (14)] in the prediction model which has been made linear using the four equilibrium points and on the coefficients $Q_i$ and $R$ for the reward function, a quadratic form controller is created using the Riccati equation (16) (refer to Appendix 2). Swing-up patterns are compared by varying the coefficient $R_i$ for the reward function and the initial values for the acrobot. At this point, the parameter for determining the shared range for each module is set to $\sigma = 1$.

### 3.2.2. Results

Figure 5 shows the trajectories for successful examples of the acrobot swing-up when $R = 0.002$. Figure 6 shows the different swing-up patterns that occur when $R$, the cost coefficient for the input to the reward function, is varied.

For instance, if link 1 represents the upper body and link 2 represents a foot, then when $R$ is varied, a posture in which the upper body is dangling ($\theta_1 = 0$, $\theta_2 = 0$) and a

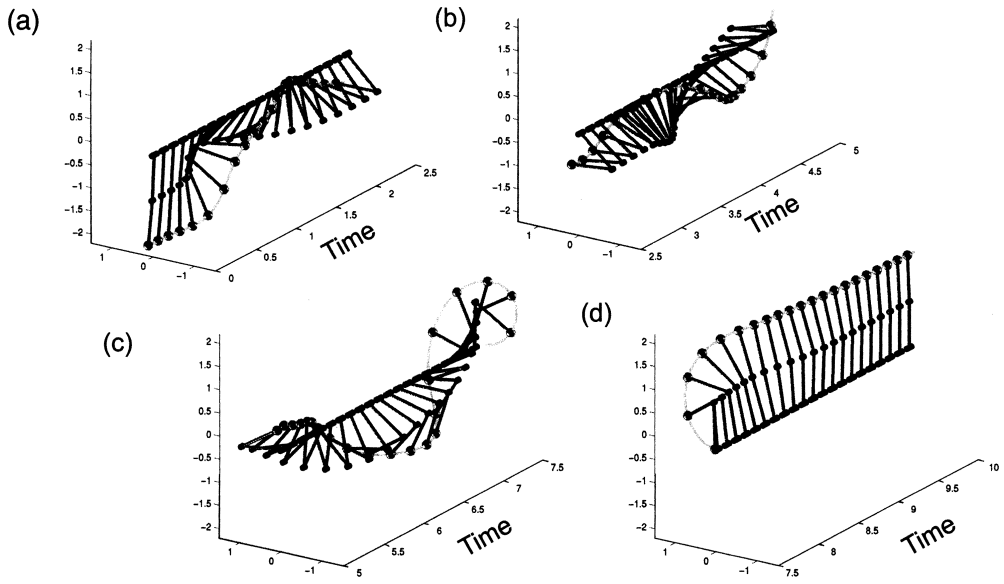


Fig. 5. An example of a successful swing-up pattern ($R = 0.002$). Each panel shows consecutive postures in (a) 0 to 2.5 s, (b) 2.5 to 5.0 s, (c) 5.0 to 7.5 s, (d) 7.5 to 10.0 s.

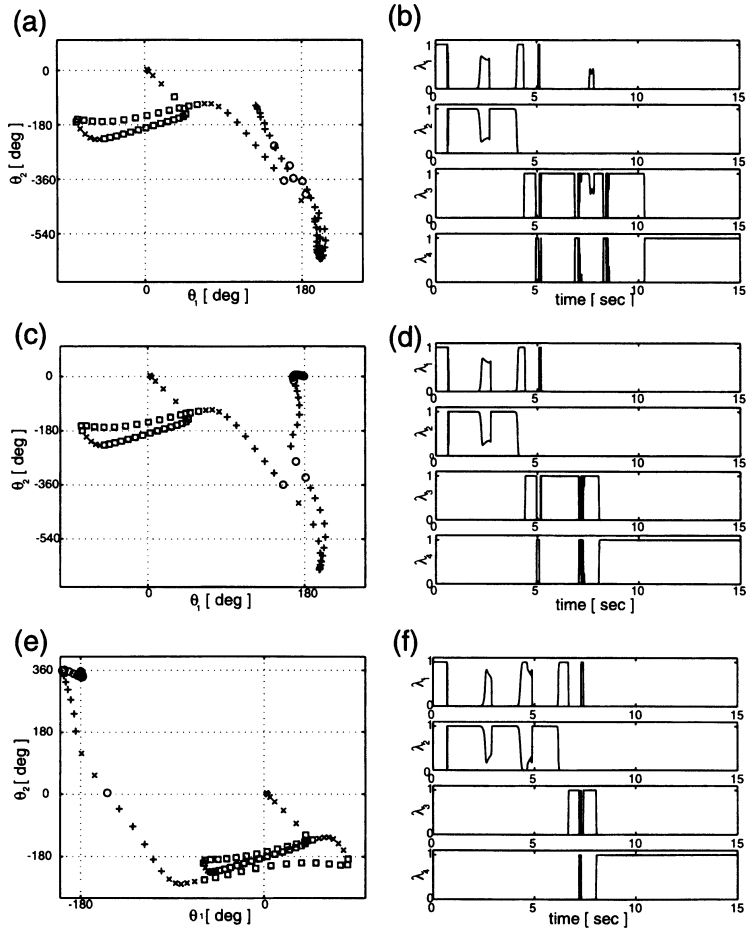


Fig. 6. Sample trajectories of the acrobot with three parameters (a, b) $R = 0.0009$, (c, d) $R = 0.001$, (e, f) $R = 0.002$. Left column (a, c, e) shows the trajectories in the $\theta_1$–$\theta_2$ plane with dominant modules; ×: module1, □: module2, +: module3, and ○ module4. Right column (b, d, f) shows the time course of the responsibility signal.

posture in which only the foot is lifted ($\theta_1 = 0$, $\theta_2 = \pi$) are alternated several times with $R = 0.0009$, 0.001, and then a posture in which only the upper body is lifted ($\theta_1 = \pi$, $\theta_2 = \pi$) is implemented [Figs. 6(a) to 6(d)], but when $R = 0.002$, the limitations on torque are considerable, and as a result there are several repetitions of transitioning between a posture in which the upper body and foot are raised ($\theta_1 = 0$, $\theta_2 = 0$) and a posture in which only the foot is raised ($\theta_1 = 0$, $\theta_2 = \pi$). The former is more frequent compared to ($R = 0.0009$, 0.001) [Figs. 6(e) and 6(f)]. Moreover, as a result of the difference in the direction of the swing-up and the number of repetitions, variations in the arrival point on the ground due to $R$ are also seen.

Furthermore, with respect to the transition for the responsibility signal $\lambda$, it is 1-2-1-2-1-3-4-1-3-4-3-4-3-4 when $R = 0.0009$ [Figs. 6(a) and 6(b)], 1-2-1-2-1-3-4-1-3-4-3-4 when $R = 0.001$ [Figs. 6(c) and 6(d)], and 1-2-1-2-1-2-1-3-4-1-3-4 when $R = 0.002$ [Figs. 6(e) and 6(f)]. Changes in the transitions for the responsibility signal $\lambda$ are also observed with respect to these different movement patterns.

When the initial values of $\theta_1$ and $\theta_2$ are varied, various different movement patterns are also acquired.

In this fashion, a nonlinear system like an acrobot can be controlled precisely using MOSAIC, and the symbolization of complex movement patterns can be achieved using the responsibility signal $\lambda$.

### 3.3. Imitation learning using movement pattern symbols

#### 3.3.1. Experimental methods

The authors used the methods of symbolization and imitation learning for the movement patterns given in Section 2.2 for the acrobot swing-up task. The movement pattern achieved by the acrobot at $R = 0.002$ is taken to be the observed movement pattern, and the authors then had the acrobot perform the swing-up task for $R = 0.002$

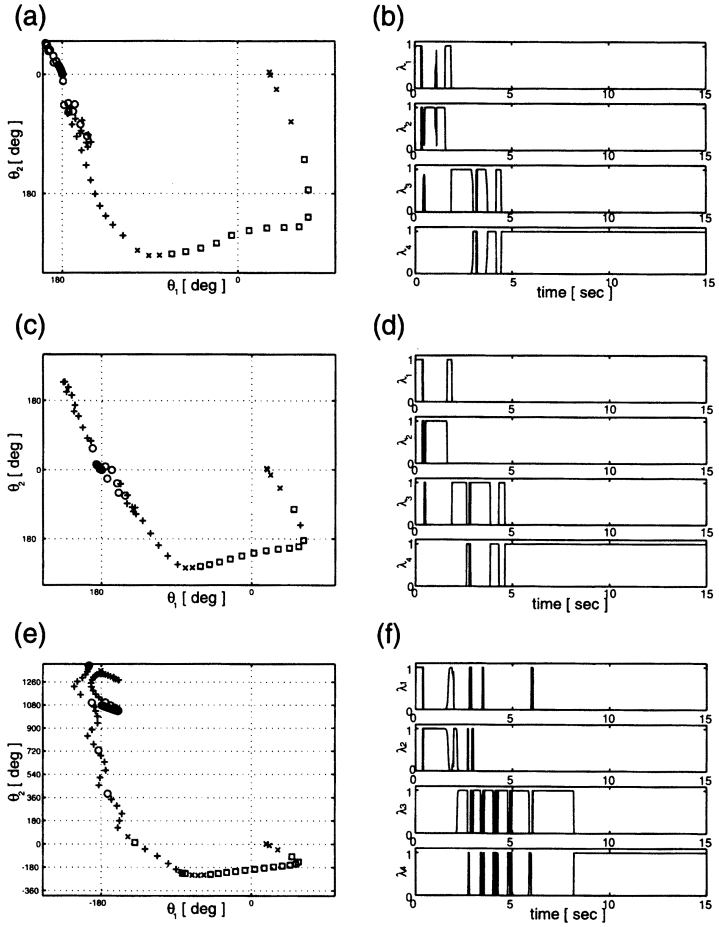


Fig. 7. Sample trajectories of the acrobot ($R = 0.002$) with (c, d) and without (e, f) the symbol imitation by observing teacher trajectory (a, b). Left column (a, c, e) shows the trajectories in the $\theta_1$–$\theta_2$ plane with dominant modules; ×: module1, □: module2, +: module3, and ○ module4. Right column (b, d, f) shows the time course of the responsibility signal.

(equivalent) and $R = 0.0035$ (slightly different). In addition, observations were made of when the movement pattern for an initial value of $\theta_1 = \pi/6$ started from $\theta_1 = \pi/12$. $\sigma_{obs} = 2$, which determines the shared range for each module, was used. Because this value is relatively large, the estimation of the responsibility signal in the module was relatively smooth. Moreover, $\varepsilon = 0.02$, a small value, was used for the parameter to average the error over a short time. As a result, switching of the responsibility signal for the observed movement pattern could be tracked.

### 3.3.2. Results

Figures 7(a) and 7(b) show the results for starting the swing-up from an initial value of $\theta_1 = \pi/6$. Swing-up was started from an initial value of $\theta_1 = \pi/12$ with this movement pattern as an example. When started from an initial value of $\theta_1 = \pi/12$ without using imitation, link 2 slowly decreased its speed while rotating and came to rest at the inversion point after link 1 swung up [Figs. 7(e), 7(f)]. When observing and imitating the movement pattern with an initial value of $\theta_1 = \pi/6$ [Figs. 7(a), 7(b)], link 2 came to rest at the inversion point without rotating after link 1 swung up. This trajectory is qualitatively different from the trajectory when imitation is not used [Figs. 7(e), 7(f)], and is more efficient due to the elimination of useless rotation, as is the case with the observed movement pattern trajectories [Figs. 7(a), 7(b)]. Given this, the new movement series can be acquired via the responsibility signal $\lambda$. Moreover, a comparison of the changes in the responsibility signal in the observed movement pattern [Fig. 7(b)] and changes in the responsibility signal predicted value [Fig. 7(d)] reveals that the transition patterns are similar, and so shows that the movement series can be recognized using the representation of the responsibility signal $\lambda$ by observing the movement pattern.

Figure 8 uses as an example the swing-up trajectory using the controller created using the cost coefficient $R = 0.001$ with respect to action output, and represents the swing-up success rate when performing imitation using seven types of controllers, $R = \{0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01\}$, and the time (average for 50 trials) required for swing-up when successful. A success is recognized when the swing-up region is reached within 30 seconds after the start of swing-up.

When $R = 0.005$ or 0.01, success is not achieved unless imitation is performed, but when imitation is performed by generating a responsibility signal from a different movement pattern, success is achieved. The success rate rises for any other parameter, meaning that imitation can provide more robust swing-up. Moreover, swing-up is successful in less time on average when imitation is performed. When $R = 0.0001$, a considerable amount of time was required for imitation, but this was because the success rate rose even though time was required.

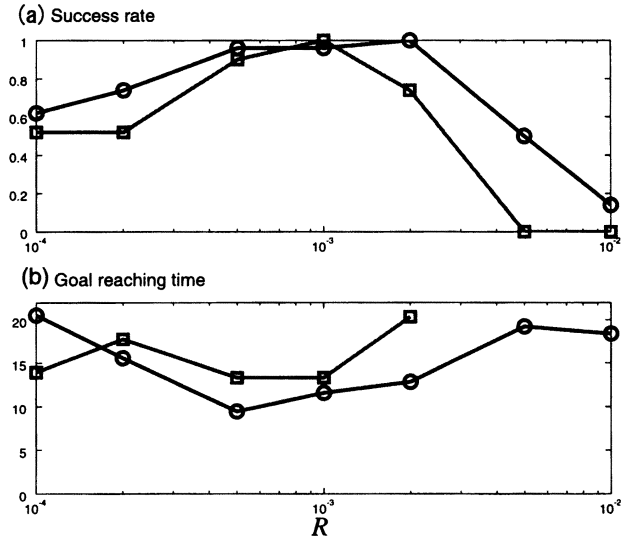


Fig. 8. (a) Success rate and (b) average goal reaching time of acrobot with (○) and without (□) symbol imitation.

## 4. Discussion

Based on the acrobot swing-up simulations, it is clear that one's own movement pattern can be improved by recognizing another's movement pattern as a symbol series using one's own prediction model and controller, and then performing "imitation" using the symbol series. In this section, the authors discuss a computational model for the brain using their proposed MOSAIC as a computational model for communication.

### 4.1. A computational model for communication

The process of communication is thought to be divided into two processes: a process for recognizing communication signals in which the internal state of another person is estimated based on voice patterns and movement patterns such as expressions and gestures provided by the other person, and a control process using communication signals in which a particular movement pattern is created using movement control and the internal state of the other person is controlled by presenting this pattern to the other person.

The imitation learning using symbols in which movement that is essentially the same as the movement pattern of another person is executed for control is slightly different

from that of the other person's and includes the two processes given above for communication. In other words, the process of recognizing another person's movement pattern as a responsibility signal series by simulating the control output using one's own prediction model is the same as the process of recognizing a communication signal. Moreover, "imitation using an estimated symbol" in which a responsibility signal obtained artificially using a movement pattern that succeeded for another person involves looking at movements that are essentially close to the other person's as a result of having the movement patterns seen in another person control one's own internal state series. This is thought to be a model for control using communication signals.

When considering language, the process of recognition is a process in which a voice signal produced continuously by another person is received through the sense of hearing, and then symbolized meaningfully at the phoneme and word level. The process of control represents a process of generating symbol strings by combining the phonemes and words above, and then generating spoken commands using one's own speech organs. Moreover, the authors believe that a similar mechanism operates in series movements for the purpose of generating/recognizing communication signals such as imitation and expression outside of language.

In this paper a symbol series is estimated from another's movement patterns, this being "imitation," and then converted to one's own movement pattern as is. This is a model of only the most basic processes in communications. The problem of hierarchical, precise action planning and grammar in language requires consideration of hierarchical symbolization and free interchange of symbol series. How this can be represented is a topic for the future, but MOSAIC can provide an effective method for symbolization which is a clue to the problems of grammar in language and of complex, precise action planning.

### 4.2. MOSAIC as a computational model of the brain

In MOSAIC, the prediction model and the controller are used as a pair. The fundamental assumption behind this computational model is that the process of generating communication is used in the process of recognizing communication. In this sense, it is based on what cognitive science calls a simulation theory [11]. Moreover, the discovery of mirror neurons [12, 13] that act both when executing movement and when observing another's movement supports MOSAIC as a computational model of the brain.

The authors believe that the production of language in people is achieved continuously based on communication using movement patterns that do not involve language, and does not assume a discontinuous evolutionary process for the basic neural structures or a computational theory for them. This idea corresponds to Broca's region, which is used when comprehending human language [14], and to the connections between the ventral premotor cortex, where mirror neurons were discovered, and the lateral portion of the cerebellum [15]. Kawato and co-workers have proposed the theory that the models necessary to convert sensation and movement in the external world in the lateral portion of the cerebellum can be acquired. Moreover, in research on brain activity using functional MRIs, the creation of a model for new tools (required for a new sensation–movement conversion) will be reported by Imamizu's group [16].

## 5. Conclusion

This paper showed that MOSAIC can adapt to highly nonlinear tasks as seen in the acrobot swing-up task. Moreover, movement patterns can be symbolized using the responsibility signal, and a new movement can be acquired by estimating its series based on the current trajectory.

Future topics include acquiring more complex movements through hierarchical combination in line with subdivided modules. In the current experiment, the authors showed that a symbol series can be estimated using the responsibility signal $\lambda$ based on a movement pattern and can be used to acquire a new movement pattern. A more autonomous hierarchical learning system could be created by acquiring the dynamics of the symbol series through learning. Moreover, if the symbolized movement series can be interchanged, then the mechanisms behind human communication and language activity may be clarified.

**Acknowledgment.** This research was performed as part of the "Research on Human and Primate Communications" through the 1998 Science and Technology Advancement Funding in the Science and Technology Agency.

## REFERENCES

1. Wolpert DM, Kawato M. Multiple paired forward and inverse models for motor control. Neural Networks 1998;11:1317–1329.
2. Haruno M, Wolpert DM, Kawato M. Mosaic: Module selection and identification for control. Neural Computation 2001;13:2201–2220.
3. Samejima K, Katagiri K, Doya K, Kawato M. Nonlinear control using reinforced learning with several predictive models. Trans IEICE 2001;J84-D-II:2092–2106.

4. Haruno M, Wolpert DM, Kawato M. Multiple paired forward–inverse models for human motor learning and control. In: Kearns MS, Salla SA, Cohn DA (editors). Advances in Neural Information Processing Systems 11. MIT Press; 1999. p 31–37.
5. Katagiri K, Doya K, Kawato M. A non-linear control method using reinforced learning with several models. Tech Rep IEICE 1998;NC98-46.
6. Doya K, Samejima K, Katagiri K, Kawato M. Multiple model-based reinforcement learning. Technical Report of Kawato Dynamic Brain Project, KDB-TR-08, 2000.
7. Spong MW. The swing up control problem for the acrobot. IEEE Control Syst 1995;15:49–55.
8. Smith MH, Lee MA, Ulieru M, Gruver WA. Design limitation of PD versus fuzzy controllers for the acrobot. International Conference on Robotics and Automation, p 1130–1135, 1997.
9. Boone G. Efficient reinforcement learning: Model-based acrobot control. Proc 1997 IEEE International Conference on Robotics and Automation, p 229–234.
10. Fleming WH, Soner MM. Controlled Markov processes and viscosity solutions. Springer-Verlag; 1993.
11. Gallese V, Goldman A. Mirror neurons and the simulation theory of mind-reading. Trends Cogn Sci 1998;2:493–501.
12. di Pellegrio G, Fadiga L, Gallese V, Rizzolatti G. Understanding motor events: A neurophysiological study. Exp Brain Res 1992;91:176–180.
13. Rizzolatti G, Gadiga L, Gallese V, Fogassi L. Premotor cortex and the recognition of motor actions. Cogn Brain Res 1996;3:131–141.
14. Rizzolatti G, Arbib MA. Language within our grasp. Trends Neurosci 1998;21:188–194.
15. Tamada T, Miyauchi S, Imamizu H, Yoshioka T, Kawato M. Cerebro-cerebellar functional connectivity related by the laterality index in tool-use learning. NeuroReport 1999;10:325–331.
16. Imamizu H, Miyauchi S, Tamada T, Sasaki Z, Takio T, Putz B, Yoshioka T, Kawato M. Human cerebellar activity reflecting an acquired internal model of a new tool. Nature 2000;403:192–195.

# APPENDIX

## 1. Equations of Motion for the Acrobot

The equations of motion for the acrobot are given in Eqs. (A.1) and (A.2):

$$d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 + h_1 + \phi_1 = 0 \tag{A.1}$$

$$d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 + h_2 + \phi_2 = T \tag{A.2}$$

$$d_{11} = m_1 r_1^2 + m_2 l_1^2 + m_2 r_2^2 + 2m_2 l_1 r_2 \cos\theta_2 + I_1 + I_2 \tag{A.3}$$

$$d_{12} = m_2 r_2^2 + m_2 l_1 r_2 \cos\theta_2 + I_2 \tag{A.4}$$

$$d_{21} = m_2 r_2^2 + m_2 l_1 r_2 \cos\theta_2 + I_2 \tag{A.5}$$

$$d_{22} = m_2 r_2^2 + I_2 \tag{A.6}$$

$$h_1 = -m_2 l_1 r_2 (2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \sin\theta_2 \tag{A.7}$$

$$h_2 = m_2 l_1 r_2 \dot{\theta}_1^2 \sin\theta_2 \tag{A.8}$$

$$\phi_1 = (m_1 r_1 + m_2 l_1) g \sin\theta_1 + m_2 r_2 g \sin\theta_1 + \theta_2 \tag{A.9}$$

$$\phi_2 = m_2 r_2 g \sin\theta_1 + \theta_2 \tag{A.10}$$

Here, $\theta_1$ and $\theta_2$ represent the angles for link 1 and link 2, $\dot{\theta}_1$ and $\dot{\theta}_2$ represent the angular velocities, $\ddot{\theta}_1$ and $\ddot{\theta}_2$ represent the angular accelerations, and $T$ represents the torque applied to link 2. The state variable is $\mathbf{x} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)'$, and the control variable is $\mathbf{u} = T$. Note that $m_1$ and $m_2$ represent the mass of links 1 and 2, $l_1$ and $l_2$, the lengths, $r_1$ and $r_2$, the distance from the joint to the center of mass of each link, and $I_1$ and $I_2$, the inertial moment of each link.

For the output torque, simulations were performed using the fixed Runge–Kutta fourth-order method, with the noise $N(0, dt)$ in a Gaussian distribution using pseudo-random numbers at the final stage input at the time $dt = 0.01$ in each simulation.

## 2. Reward Model Coefficient Matrices and LQC Creation

$Q_i$ at each equilibrium point used here is given by

$$Q_1 = \begin{bmatrix} -1 & -1/4 & 0 & 0 \\ -1/4 & -1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.11}$$

$$Q_2 = \begin{bmatrix} -1/2 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.12}$$

$$Q_3 = \begin{bmatrix} 1/2 & -1/4 & 0 & 0 \\ -1/4 & -1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.13}$$

$$Q_4 = \begin{bmatrix} 1 & 1/4 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{(A.14)}$$

Here, when the Riccati equation (16) is solved, a stable answer is not obtained unless each element is a positive number. Thus, when $Q$ has negative elements, in other words when the reward for a joint is rising with respect to the direction of instability, first the sign for the element in $Q$ for the destabilizing joint ($\theta_1$ and $\theta_2$ for $Q_1$, $\theta_2$ for $Q_2$, $\theta_2$ for $Q_3$), and then the LQC gain is found. An unstable controller is then created by inverting the sign for each angle component and velocity component for the joint which inverts $Q$ from among the resulting components in the gain $K$.

## **AUTHORS** (from left to right)

**Kazuyuki Samejima** graduated from the Department of Electronics and Informatics at Tokyo University of Agriculture and Technology in 1993, completed his doctoral studies in 1999, and became a researcher with the Computational Neurophysiology Group with the Kawato Learning and Dynamic Brain Project, Japan Science and Technology Corp., where he worked on research on reinforcement learning, basal ganglion models, and computational theory. He holds a D.Eng. degree, and is a member of the Society for Neuroscience and the Japanese Neural Network Society.

**Ken'ichi Katagiri** graduated from Department of Applied Animal Science at Hokkaido University in 1997, completed his master's course at Nara Institute of Science and Technology in 1999, and joined Neore Japan. While working on his master's, he did research on reinforcement learning and optimization control theory.

**Kenji Doya** (member) completed his master's course in statistics and engineering at the University of Tokyo in 1984 and became a lecturer there. In 1991 he became a researcher with the Department of Biological Sciences at the University of California, San Francisco. In 1994 he became a lead researcher with the Human Information and Communications Laboratory at ATR. In 1995 he was appointed an assistant professor with the Graduate School at Nara Institute of Science and Technology. In 1996 he became a group leader with the Computational Neurophysiology Group with the Kawato Learning and Dynamic Brain Project, Japan Science and Technology Corp. In 1999 he became a CREST research representative for the Japan Science and Technology Corp. at the International Electrical Communications Laboratory. He is a member of the Society for Neuroscience, International Neural Network Society, and the Japanese Neural Network Society.

**Mituo Kawato** (member) graduated from the Department of Physics at the University of Tokyo in 1976, completed his doctoral studies in engineering in 1981, and became a lecturer at the University of Osaka. In 1988 he joined the ATR Audiovisual Perception Laboratory. From 1992 through 2001 he was head of Laboratory 3 at the Human Information and Communications Laboratory at ATR. Since 2000 he has been a project leader with the Cyber-Human Project in the Computational Neuroscience Project at ATR. From 2002 through 2005 he was a visiting professor with the Electronics Science Laboratory at the University of Hokkaido, a visiting professor at the University of Genoa in Italy in 2003, a visiting professor at the Kanazawa Institute of Technology since 2004, and leader of the Kawato Learning and Dynamic Brain Project, Japan Science and Technology Corp. Since 2000 he has been a visiting professor at the Graduate School at Nara Institute of Science and Technology, and has been pursuing research on computational neuroscience. He has received the Yonezawa Prize, the University of Osaka Science Award, the Science and Technology Agency's Secretary's Award, and the Tsukawara Prize. His books include *The Framework of the Brain* and *A Computational Theory of the Brain*. He is the editor of the journal *Neural Networks*, and is a member of the board of the International Neural Network Society and of the editorial board of the Japanese Neural Network Society. He is also a member of the Executive Committee of the International Association for the Study of Attention and Performance.