LETTER **Communicated by Andrew Barto**

Multiple Model-Based Reinforcement Learning

Kenji Doya
doya@atr.co.jp
Human Information Science Laboratories, ATR International, Seika, Soraku, Kyoto 619-0288, Japan; CREST, Japan Science and Technology Corporation, Seika, Soraku, Kyoto 619-0288, Japan; Kawato Dynamic Brain Project, ERATO, Japan Science and Technology Corporation, Seika, Soraku, Kyoto 619-0288, Japan; and Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan

Kazuyuki Samejima
samejima@atr.co.jp
Human Information Science Laboratories, ATR International, Seika, Soraku, Kyoto 619-0288, Japan, and Kawato Dynamic Brain Project, ERATO, Japan Science and Technology Corporation, Seika, Soraku, Kyoto 619-0288, Japan

Ken-ichi Katagiri
keniti-k@syd.odn.ne.jp
ATR Human Information Processing Research Laboratories, Seika, Soraku, Kyoto 619-0288, Japan, and Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan

Mitsuo Kawato
kawato@atr.co.jp
Human Information Science Laboratories, ATR International, Seika, Soraku, Kyoto 619-0288, Japan; Kawato Dynamic Brain Project, ERATO, Japan Science and Technology Corporation, Seika, Soraku, Kyoto 619-0288, Japan; and Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan

We propose a modular reinforcement learning architecture for nonlinear, nonstationary control tasks, which we call multiple model-based reinforcement learning (MMRL). The basic idea is to decompose a complex task into multiple domains in space and time based on the predictability of the environmental dynamics. The system is composed of multiple modules, each of which consists of a state prediction model and a reinforcement learning controller. The "responsibility signal," which is given by the softmax function of the prediction errors, is used to weight the outputs of multiple modules, as well as to gate the learning of the prediction models and the reinforcement learning controllers. We formulate MMRL for both discrete-time, finite-state case and continuous-time, continuous-state case. The performance of MMRL was demonstrated for discrete case

in a nonstationary hunting task in a grid world and for continuous case in a nonlinear, nonstationary control task of swinging up a pendulum with variable physical parameters.

1 Introduction

A big issue in the application of reinforcement learning (RL) to real-world control problems is how to deal with nonlinearity and nonstationarity. For a nonlinear, high-dimensional system, the conventional discretizing approach necessitates a huge number of states, which makes learning very slow. Standard RL algorithms can perform badly when the environment is nonstationary or has hidden states. These problems have motivated the introduction of modular or hierarchical RL architectures (Singh, 1992; Dayan & Hinton, 1993; Littman, Cassandra, & Kaelbling, 1995; Wiering & Schmidhuber, 1998; Parr & Russel, 1998; Sutton, Precup, & Singh, 1999; Morimoto & Doya, 2001). The basic problem in modular or hierarchical RL is how to decompose a complex task into simpler subtasks.

This article presents a new RL architecture based on multiple modules, each composed of a state prediction model and an RL controller. With this architecture, a nonlinear or nonstationary control task, or both, is decomposed in space and time based on the local predictability of the environmental dynamics.

The mixture of experts architecture (Jacobs, Jordan, Nowlan, & Hinton, 1991) has been applied to nonlinear or nonstationary control tasks (Gomi & Kawato, 1993; Cacciatore & Nowlan, 1994). However, the success of such modular architecture depends strongly on the capability of the gating network to decide which of the given modules should be recruited at any particular moment.

An alternative approach is to provide each of the experts with a prediction model of the environment and to use the prediction errors for selecting the controllers. In Narendra, Balakrishnan, and Ciliz (1995), the model that makes the smallest prediction error among a fixed set of prediction models is selected, and its associated single controller is used for control. However, when the prediction models are to be trained with little prior knowledge, task decomposition is initially far from optimal. Thus, the use of "hard" competition can lead to suboptimal task decomposition.

Based on the Bayesian statistical framework, Pawelzik, Kohlmorge, and Müller (1996) proposed the use of annealing in a "soft" competition network for time-series prediction and segmentation. Tani and Nolfi (1999) used a similar mechanism for hierarchical sequence prediction. The use of the softmax function for module selection and combination was originally proposed for a tracking control paradigm as the multiple paired forward-inverse models (MPFIM) (Wolpert & Kawato, 1998; Wolpert, Miall, & Kawato, 1998; Haruno, Wolpert, & Kawato, 1999). It was recently

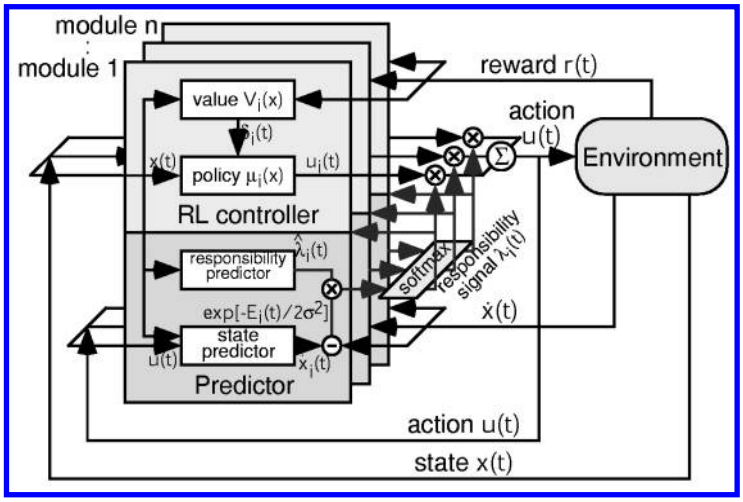

Figure 1: Schematic diagram of the MMRL architecture.

reformulated as modular selection and identification for control (MOSAIC) (Wolpert & Ghahramani, 2000; Haruno, Wolpert, & Kawato, 2001).

In this article, we apply the idea of a softmax selection of modules to the paradigm of reinforcement learning. The resulting learning architecture, which we call multiple model-based reinforcement learning (MMRL), learns to decompose a nonlinear or nonstationary task through the competition and cooperation of multiple prediction models and reinforcement learning controllers.

In section 2, we formulate the basic MMRL architecture and in section 3 describe its implementation in discrete-time and continuous-time cases, including multiple linear quadratic controllers (MLQC). We first test the performance of the MMRL architecture for the discrete case in a hunting task with multiple preys in a grid world (section 4). We also demonstrate the performance of MMRL for continuous case in a nonlinear, nonstationary control task of swinging up a pendulum with variable physical parameters (section 5).

2 Multiple Model-Based Reinforcement Learning

Figure 1 shows the overall organization of the MMRL architecture. It is composed of n modules, each of which consists of a state prediction model and a reinforcement learning controller.

The basic idea of this modular architecture is to decompose a nonlinear or nonstationary task into multiple domains in space and time so that within each of the domains, the environmental dynamics is predictable. The

action output of the RL controllers, as well as the learning rates of both the predictors and the controllers, are weighted by the "responsibility signal," which is a gaussian softmax function of the errors in the outputs of the prediction models. The advantage of this module selection mechanism is that the areas of specialization of the modules are determined in a bottom-up fashion based on the nature of the environment. Furthermore, for each area of module specialization, the design of the control strategy is facilitated by the availability of the local model of the environmental dynamics.

In the following, we consider a discrete-time, finite-state environment,

$$P(x(t) \mid x(t-1), u(t-1)) = F(x(t), x(t-1), u(t-1)), \quad (t = 1, 2, \ldots), \tag{2.1}$$

where $x \in \{1, \ldots, N\}$ and $u \in \{1, \ldots, M\}$ are discrete states and actions, and a continuous-time, continuous-state environment,

$$\dot{x}(t) = f(x(t), u(t)) + \nu(t), \qquad (t \in [0, \infty)), \tag{2.2}$$

where $x \in R^N$ and $u \in R^M$ are state and action vectors, and $\nu \in R^N$ is noise. Actions are given by a policy—either a stochastic one,

$$P(u(t) \mid x(t)) = G(u(t), x(t)), \tag{2.3}$$

or a deterministic one,

$$u(t) = g(x(t)). \tag{2.4}$$

The reward $r(t)$ is given as a function of the state $x(t)$ and the action $u(t)$. The goal of reinforcement learning is to improve the policy so that more rewards are acquired in the long run. The basic strategy of reinforcement learning is to estimate cumulative future reward under the current policy as the "value function" $V(x)$ for each state and then to improve the policy based on the value function. We define the value function of the state $x(t)$ under the current policy as

$$V(x(t)) = E\left[\sum_{k=0}^{\infty} \gamma^k r(t+k)\right] \tag{2.5}$$

in discrete case (Sutton & Barto, 1998) and

$$V(x(t)) = E\left[\int_0^{\infty} e^{-\frac{s}{\tau}} r(t+s)\, ds\right] \tag{2.6}$$

in continuous case (Doya, 2000), where $0 \le \gamma \le 1$ and $0 < \tau$ are the parameters for discounting future reward.

2.1 Responsibility Signal. The purpose of the prediction model in each module is to predict the next state (discrete time) or the temporal derivative of the state (continuous time) based on the observation of the state and the action. The responsibility signal $\lambda_i(t)$ (Wolpert & Kawato, 1998; Haruno et al., 1999, 2001) is given by the relative goodness of predictions of multiple prediction models.

For a unified description, we denote the new state in the discrete case as

$$y(t) = x(t) \tag{2.7}$$

and the temporal derivative of the state in the continuous case as

$$y(t) = \dot{x}(t). \tag{2.8}$$

The basic formula for the responsibility signal is given by Bayes' rule,

$$\lambda_i(t) = P(i \mid y(t)) = \frac{P(i)P(y(t) \mid i)}{\sum_{j=1}^{n} P(j)P(y(t) \mid j)}, \tag{2.9}$$

where $P(i)$ is the prior probability of selecting module i and $P(y(t) \mid i)$ is the likelihood of model i given the observation $y(t)$.

In the discrete case, the prediction model gives the probability distribution of the new state $\hat{x}(t)$ based on the previous state $x(t-1)$ and the action $u(t-1)$ as

$$P(\hat{x}(t) \mid x(t-1), u(t-1)) = F_i(\hat{x}(t), x(t-1), u(t-1)) \quad (i = 1, \ldots, n). \tag{2.10}$$

If there is no prior knowledge of module selection, we take the priors as uniform ($P(i) = 1/n$), and then the responsibility signal is given by

$$\lambda_i(t) = \frac{F_i(x(t), x(t-1), u(t-1))}{\sum_{j=1}^{n} F_j(x(t), x(t-1), u(t-1))}, \tag{2.11}$$

where $x(t)$ is the newly observed state.

In the continuous case, the prediction model gives the temporal derivative of the state:

$$\hat{\dot{x}}_i(t) = f_i(x(t), u(t)). \tag{2.12}$$

By assuming that the prediction error is gaussian with variance σ^2, the responsibility signal is given by the gaussian softmax function,

$$\lambda_i(t) = \frac{e^{-\frac{1}{2\sigma^2}\|\dot{x}(t) - \hat{\dot{x}}_i(t)\|^2}}{\sum_{j=1}^{n} e^{-\frac{1}{2\sigma^2}\|\dot{x}(t) - \hat{\dot{x}}_i(t)\|^2}}, \tag{2.13}$$

where $\dot{x}(t)$ is the observed state change.

2.2 Module Weighting by Responsibility Signal. In the MMRL architecture, the responsibility signal $\lambda_i(t)$ is used for four purposes: weighting the state prediction outputs, gating the learning of prediction models, weighting the action outputs, and gating the learning of reinforcement learning controller:

- **State prediction:** The outputs of the prediction models are weighted by the responsibility signal $\lambda_i(t)$. In the discrete case, the prediction of the next state is given by

$$P(\hat{x}(t)) = \sum_{i=1}^{n} \lambda_i(t) F_i(\hat{x}(t), x(t-1), u(t-1)). \tag{2.14}$$

  In the continuous case, the predicted state derivative is given by

$$\hat{\dot{x}}(t) = \sum_{i=1}^{n} \lambda_i(t) \hat{\dot{x}}_i(t). \tag{2.15}$$

  These predictions are used in model-based RL algorithms and also for the annealing of σ, as described later.

- **Prediction model learning:** The responsibility signal $\lambda_i(t)$ is also used for weighting the parameter update of the prediction models. In general, it is realized by scaling the error signal of prediction model learning by $\lambda_i(t)$.

- **Action output:** The outputs of reinforcement learning controllers are linearly weighted by $\lambda_i(t)$ to make the action output. In the discrete case, the probability of taking an action $u(t)$ is given by

$$P(u(t)) = \sum_{i=1}^{n} \lambda_i(t) G_i(u(t), x(t)). \tag{2.16}$$

  In the continuous case, the output is given by the interpolation of modular outputs

$$u(t) = \sum_{i=1}^{n} \lambda_i(t) u_i(t) = \sum_{i=1}^{n} \lambda_i(t) g_i(x(t)). \tag{2.17}$$

- **Reinforcement learning:** $\lambda_i(t)$ is also used for weighting the learning of the RL controllers. The actual equation for the parameter update varies with the choice of the RL algorithms, which are detailed in the next section. When a temporal difference (TD) algorithm (Barto, Sutton, & Anderson, 1983; Sutton, 1988; Doya, 2000) is used, the TD error,

$$\delta(t) = r(t) + \gamma V(x(t+1)) - V(x(t)), \tag{2.18}$$

in the discrete case and

$$\delta(t) = \hat{r}(t) - \frac{1}{\tau} V(t) + \dot{V}(t) \tag{2.19}$$

in the continuous case, is weighted by the responsibility signal

$$\delta_i(t) = \lambda_i(t)\delta(t) \tag{2.20}$$

for learning of the ith RL controller.

Using the same weighting factor $\lambda_i(t)$ for training the prediction models and the RL controllers helps each RL controller learn an appropriate policy and its value function for the context under which its paired prediction model makes valid predictions.

2.3 Responsibility Predictors. When there is some prior knowledge or belief about module selection, we incorporate the "responsibility predictors" (Wolpert & Kawato, 1998; Haruno et al., 1999, 2001). By assuming that their outputs $\hat{\lambda}_i(t)$ are proportional to the prior probability of module selection, from equation 2.9, the responsibility signal is given by

$$\lambda_i(t) = \frac{\hat{\lambda}_i(t)P(y(t) \mid i)}{\sum_{j=1}^{n} \hat{\lambda}_j(t)P(y(t) \mid j)}. \tag{2.21}$$

In modular decomposition of a task, it is desired that modules do not switch too frequently. This can be enforced by incorporating responsibility priors based on the assumption of temporal continuity and spatial locality of module activation.

2.3.1 Temporal Continuity. The continuity of module selection is incorporated by taking the previous responsibility signal as the responsibility prediction signal. In the discrete case, we take the responsibility prediction based on the previous responsibility,

$$\hat{\lambda}_i(t) = \lambda_i(t-1)^{\alpha}, \tag{2.22}$$

where $0 < \alpha < 1$ is a parameter that controls the strength of the memory effect. From equations 2.21 and 2.22, the responsibility signal at time t is given by the product of likelihoods of past module selection,

$$\lambda_i(t) = \frac{1}{Z(t)} \prod_{k=0}^{t} P(x(t-k) \mid i)^{\alpha^k}, \tag{2.23}$$

where $Z(t)$ denotes the normalizing factor, that is, $Z(t) = \sum_{j=1}^{n} \prod_{k=0}^{t} P(x(t-k) \mid j)^{\alpha^k}$.

In the continuous case, we choose the prior

$$\hat{\lambda}_i(t) = \lambda_i(t - \Delta t)^{\Delta t \alpha^{\Delta t}}, \tag{2.24}$$

where Δt is an arbitrarily small time difference (note 2.24 coincides with 2.22 with $\Delta t = 1$).

Since the likelihood of the module i is given by the gaussian $P(\dot{x}(t) \mid i) = e^{-\frac{1}{2\sigma^2}\|\dot{x}(t) - \hat{\dot{x}}_i(t)\|^2}$, from recursion as in equation 2.23, the responsibility signal at time t is given by

$$\begin{aligned}\lambda_i(t) &= \frac{1}{Z(t)} \prod_{k=0}^{t/\Delta t} P(\dot{x}(t - k\Delta t) \mid i)^{\Delta t \alpha^{k\Delta t}} \\ &= \frac{1}{Z(t)} e^{-\frac{1}{2\sigma^2}\Delta t \sum_{k=0}^{t/\Delta t} \|\dot{x}(t-k\Delta t) - \hat{\dot{x}}_i(t-k\Delta t)\|^2 \alpha^{k\Delta t}},\end{aligned} \tag{2.25}$$

that is, a gaussian softmax function of temporally weighted squared errors. In the limit of $\Delta t \to 0$, equation 2.25 can be represented as

$$\lambda_i(t) = \frac{e^{-\frac{1}{2\sigma^2}E_i(t)}}{\sum_{j=1}^{n} e^{-\frac{1}{2\sigma^2}E_j(t)}}, \tag{2.26}$$

where $E_i(t)$ is a low-pass filtered prediction error

$$\dot{E}_i(t) = \log \alpha \, E_i(t) + \|\dot{x}(t) - \hat{\dot{x}}_i(t)\|^2. \tag{2.27}$$

The use of this low-pass filtered prediction error for responsibility prediction is helpful in avoiding chattering of the responsibility signal (Pawelzik et al., 1996).

2.3.2 Spatial Locality. In the continuous case, we consider a gaussian spatial prior,

$$\hat{\lambda}_i(t) = \frac{e^{-\frac{1}{2}(x(t)-\mathbf{c}_i)' M_i^{-1} (x(t)-\mathbf{c}_i)}}{\sum_{j=1}^{n} e^{-\frac{1}{2}(x(t)-\mathbf{c}_j)' M_j^{-1} (x(t)-\mathbf{c}_j)}}, \tag{2.28}$$

where $\mathbf{c}_i$ is the center of the area of specialization, M_i is a covariance matrix that specifies the shape, and $'$ denotes transpose. These parameters are updated so that they approximate the distribution of the input state $x(t)$ weighted by the responsibility signal,

$$\dot{\mathbf{c}}_i = \eta_c \lambda_i(t)(-\mathbf{c}_i + x(t)), \tag{2.29}$$

$$\dot{M}_i = \eta_M \lambda_i(t)[-M_i + (x(t) - \mathbf{c}_i)(x(t) - \mathbf{c}_i)'], \tag{2.30}$$

where η_c and η_M are update rates.

3 Implementation of MMRL Architecture

For the RL controllers of MMRL, it is generally possible to use model-free RL algorithms, such as actor-critic and Q-learning. However, because the prediction models of the environmental dynamics are intrinsic components of the architecture, it is advantageous to use these prediction models not just for module selection but also for designing RL controllers. In the following, we describe the use of model-based RL algorithms for discrete-time and continuous-time cases. One special implementation for the continuous-time case is the use of multiple linear quadratic controllers derived from linear dynamic models and quadratic reward models.

3.1 Discrete-Time MMRL. Now we consider implementation of the MMRL architecture for discrete-time, finite-state, and finite-action problems. The standard way of using a predictive model in RL is to use it for action selection by the one-step search,

$$u(t) = \arg\max_u E[\hat{r}(x(t), u) + \gamma V(\hat{x}(t+1))], \tag{3.1}$$

where $\hat{r}(x(t), u)$ is the predicted immediate reward and $\hat{x}(t+1)$ is the next state predicted from the current state $x(t)$ and a candidate action u.

In order to implement this algorithm, we provide each module with a reward model $\hat{r}_i(x, u)$, a value function $V_i(x)$, and a dynamic model $F_i(\hat{x}, x, u)$. Each candidate action u is then evaluated by

$$\begin{aligned} q(x(t), u) &= E[\hat{r}(x(t), u) + \gamma V(\hat{x}(t+1)) \mid u] \\ &= \sum_{i=1}^{n} \lambda_i(t)[\hat{r}_i(x(t), u) + \gamma \sum_{\hat{x}=1}^{N} V_i(\hat{x}) F_i(\hat{x}, x(t), u)]]. \end{aligned} \tag{3.2}$$

For the sake of exploration, we use a stochastic version of the greedy action selection, equation 3.1, where the action $u(t)$ is selected by a Gibbs distribution,

$$P(u \mid x(t)) = \frac{e^{\beta q(x(t),u)}}{\sum_{u'=1}^{M} e^{\beta q(x(t),u')}}, \tag{3.3}$$

where β controls the stochasticity of action selection.

The parameters are updated by the error signals weighted by the responsibility signal: $\lambda_i(t)(F_i(j, x(t-1), u(t-1)) - c(j, x(t)))$ for the dynamic model ($j = 1, \ldots, N$; $c(j, x) = 1$ if $j = x$ and zero otherwise), $\lambda_i(t)(\hat{r}_i(x(t), u(t)) - r(t))$ for the reward model, and $\lambda_i(t)\delta(t)$ for the value function model.

3.2 Continuous-Time MMRL. Next we consider a continuous-time MMRL architecture. A model-based RL algorithm for a continuous-time,

continuous-state system (see equation 2.2) is derived from the Hamilton-Jacobi-Bellman (HJB) equation,

$$\frac{1}{\tau}V(x(t)) = \max_u \left[r(x(t), u) + \frac{\partial V(x(t))}{\partial x} f(x(t), u) \right], \tag{3.4}$$

where τ is the time constant of reward discount (Doya, 2000). Under the assumptions that the system is linear with respect to the action and the action cost is convex, a greedy policy is given by

$$u = g\left(\frac{\partial f(x, u)}{\partial u}' \frac{\partial V(x)}{\partial x}' \right), \tag{3.5}$$

where $\frac{\partial V(x)}{\partial x}'$ is a vector representing the steepest ascent direction of the value function, $\frac{\partial f(x,u)}{\partial u}'$ is a matrix representing the input gain of the dynamics, and g is a sigmoid function whose shape is determined by the control cost (Doya, 2000).

To implement the HJB-based algorithm, we provide each module with a dynamic model $f_i(x, u)$ and a value model $V_i(x)$. The outputs of the dynamic models, equation 2.12, are compared with the actually observed state dynamics $\dot{x}(t)$ to calculate the responsibility signal $\lambda_i(t)$ according to equation 2.13.

The model outputs are linearly weighted by $\lambda_i(t)$ for state prediction,

$$\hat{\dot{x}}(t) = \sum_{i=1}^{n} \lambda_i(t) f_i(x(t), u(t)), \tag{3.6}$$

and value function estimation,

$$V(x) = \sum_{i=1}^{n} \lambda_i(t) V_i(x). \tag{3.7}$$

The derivatives of the dynamic models $\frac{\partial f_i(x,u)}{\partial u}$ and value models $\frac{\partial V_i(x)}{\partial x}$ are used to calculate the action for each module:

$$u_i(t) = g\left(\frac{\partial f_i(x, u)}{\partial u}' \frac{\partial V_i(x)}{\partial x}' \right)\Big|_{x(t)}. \tag{3.8}$$

They are then weighted by $\lambda_i(t)$ according to equation 2.17 to make the actual action $u(t)$.

Learning is based on the weighted prediction errors $\lambda_i(t)(\hat{\dot{x}}_i(t) - \dot{x}(t))$ for dynamic models and $\lambda_i(t)\delta(t)$ for value function models.

3.3 Multiple Linear Quadratic Controllers. In a modular architecture like the MMRL, the use of universal nonlinear function approximators with large numbers of degrees of freedom can be problematic because it can lead to an undesired solution in which a single module tries to handle most of the task domain. The use of linear models for the prediction models and the controllers is a reasonable choice because local linear models have been shown to have good properties of quick learning and good generalization (Schaal & Atkeson, 1996). Furthermore, if the reward function is locally approximated by a quadratic function, then we can use a linear quadratic controller (see, e.g., Bertsekas, 1995) for the RL controller design.

We use a local linear dynamic model,

$$\hat{\dot{x}}_i(t) = A_i(x(t) - x_i^d) + B_i u(t), \tag{3.9}$$

and a local quadratic reward model,

$$\hat{r}_i(x(t), u(t)) = r_i^0 - \frac{1}{2}(x(t) - x_i^r)' Q_i (x(t) - x_i^r) - \frac{1}{2} u'(t) R_i u(t), \tag{3.10}$$

for each module, where x_i^d, x_i^r is the center of local prediction for state and reward, respectively. The r_i^0 is a bias of quadratic reward model.

The value function is given by the quadratic form,

$$V_i(x) = v_i^0 - \frac{1}{2}(x - x_i^v)' P_i (x - x_i^v). \tag{3.11}$$

The matrix P_i is given by solving the Riccati equation,

$$0 = \frac{1}{\tau} P_i - P_i A_i - A_i' P_i + P_i B_i R_i^{-1} B_i' P_i - Q_i. \tag{3.12}$$

The center x_i^v and the bias v_i^0 of the value function are given by

$$x_i^v = (Q_i + P_i A_i)^{-1} (Q_i x_i^r + P_i A_i x_i^d), \tag{3.13}$$

$$\frac{1}{\tau} v_i^0 = r_i^0 - \frac{1}{2}(x_i^v - x_i^r)' Q_i (x_i^v - x_i^r). \tag{3.14}$$

Then the optimal feedback control for each module is given by the linear feedback,

$$u_i(t) = -R_i^{-1} B_i^T P_i (x(t) - x_i^v). \tag{3.15}$$

The action output is given by weighting these controller outputs by the responsibility signal $\lambda_i(t)$:

$$u(t) = \sum_{i=1}^{n} \lambda_i(t) u_i(t). \tag{3.16}$$

The parameters of the local linear models A_i, B_i, and x_i^d and those of the quadratic reward models r_i^0, Q_i, and R_i are updated by the weighted prediction errors $\lambda_i(t)(\hat{\dot{x}}_i(t) - \dot{x}(t))$ and $\lambda_i(t)(\hat{r}_i(x, u) - r(t))$, respectively. When we assume that the update of these models is slow enough, then the Riccati equations, 3.12, may be recalculated only intermittently. We call this method multiple linear quadratic controllers (MLQC).

4 Simulation: Discrete Case

In order to test the effectiveness of the MMRL architecture, we first applied the discrete MMRL architecture to a nonstationary hunting task in a grid world. The hunter agent tries to catch a prey in a 7×7 torus grid world. There are 47 states representing the position of the prey relative to the hunter. The hunter chooses one of five possible actions: {north (N), east (E), south (S), west (W), stay}. A prey moves in a fixed direction during a trial. At the beginning of each trial, one of four movement directions {NE, NW, SE, SW} is randomly selected, and a prey is placed at a random position in the grid world. When the hunter catches the prey by stepping into the same grid with the prey, a reward $r(t) = 10$ is given. Each step of movement costs $r(t) = -1$. A trial is terminated when the hunter catches a prey or fails to catch it within 100 steps.

In order to compare the performance of MMRL with conventional methods, we applied standard Q-learning and compositional Q-learning (CQ-L) (Singh, 1992) to the same task. A major difference between CQ-L and MMRL is the criterion for modular decomposition: CQ-L uses the consistency of the modular value functions, while MMRL uses the prediction errors of dynamic models. In CQ-L, the gating network as well as component Q-learning modules are trained so that the composite Q-value well approximates the action value function of the entire problem. In the original CQ-L (Singh, 1992), the output of the gating network was based on the "augmenting bit" that explicitly signaled the change in the context. Since our goal now is to let the agent learn appropriate decomposition of the task without an explicit cue, we used a modified CQ-L (see the appendix for the details of the algorithm and the parameters).

4.1 Results. Figure 2 shows the performance difference of standard Q-learning, CQ-L, and MMRL in the hunting task. The modified CQ-L did not perform significantly better than standard, flat Q-learning. Investigation of the modular Q functions of CQ-L revealed that in most simulation runs, modules did not appropriately differentiate for four different kinds of preys. On the other hand, the performance of MMRL approached close to theoretical optimum. This was because four modules successfully specialized in one of four kinds of prey movement.

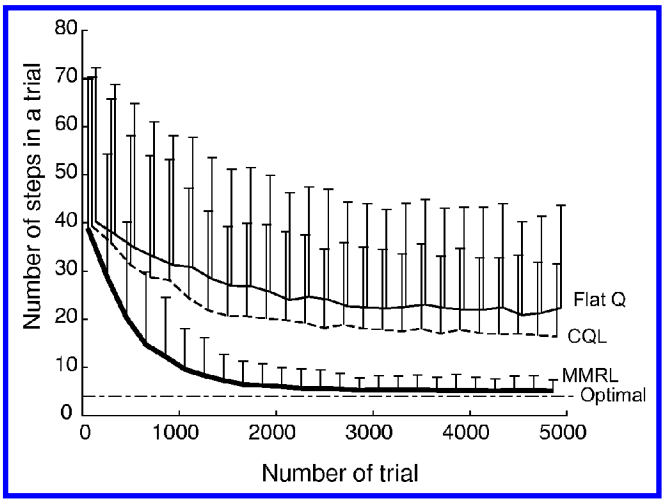

Figure 2: Comparison of the performance of standard Q-learning (gray line), modified CQ-L (dashed line), and MMRL (thick line) in the hunting task. The average number of steps needed for catching a prey during 200 trial epochs in 10 simulation runs is plotted. The dash-dotted line shows the theoretically minimal average steps required for catching the prey.

Figure 3 shows examples of the value functions and the prediction models learned by MMRL. From the output of the prediction models F_i, it can be seen that the modules 1, 2, 3, and 4 were specialized for the prey moving to NE, NW, SW, and SE, respectively. The landscapes of the value functions $V_i(x)$ are in accordance with these movement directions of the prey.

A possible reason for the difference in the performance of CQ-L and MMRL in this task is the difficulty of module selection. In CQ-L, when the prey is far from the hunter, the differences in discounted Q values for different kinds of prey are minor. Thus, it would be difficult to differentiate modules based solely on the Q values. In MMRL, on the other hand, module selection based on the state change, in this case prey movement, is relatively easy even when the prey is far from the hunter.

5 Simulation: Continuous Case

In order to test the effectiveness of the MMRL architecture for control, we applied the MLQC algorithm described in section 3.3 to the task of swinging up a pendulum with limited torque (see Figure 4) (Doya, 2000). The driving torque T is limited in $[-T^{\max}, T^{\max}]$ with $T^{\max} < mgl$. The pendulum has to be swung back and forth at the bottom to build up enough momentum for a successful swing up.

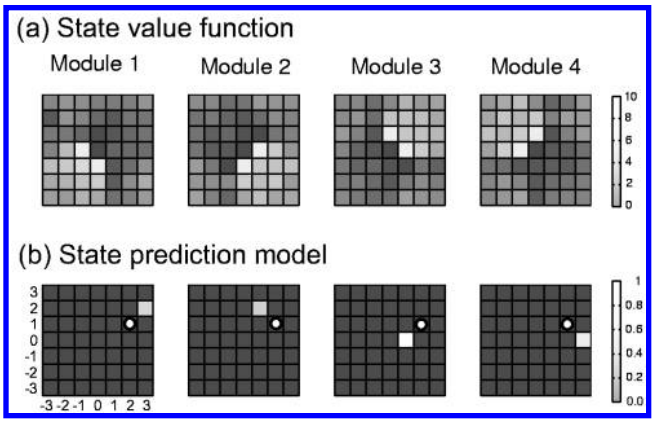

Figure 3: Example of value functions and prediction models learned by MMRL after 10,000 trials. Each slot in the grid shows the position of the prey relative to the hunter, which was used as the state x. (a) The state value functions $V_i(x)$. (b) The prediction model outputs $F_i(\hat{x}, x, u)$, where the current state x of the prey was fixed as $(2, 1)$, shown by the circle, and the action u was fixed as "stay."

The state space was two-dimensional: $x = (\theta, \dot{\theta})' \in [-\pi, \pi] \times \mathbf{R}$, where θ is the joint angle ($\theta = 0$ means the pendulum hanging down). The action was $u = T$. The reward was given by the height of the tip and the negative squared torque:

$$r(x, u) = -\cos\theta - \frac{1}{2}RT^2. \tag{5.1}$$

A trial was started from random joint angle $\theta \in [-\pi/4, \pi/4]$ with no angular velocity. We devised the following automatic annealing process for the parameter σ of the softmax function for the responsibility signal, equation 2.26,

$$\sigma_{k+1} = \eta a E_k + (1 - \eta)\sigma_k, \tag{5.2}$$

where k denotes the number of trial and E_k is the average state prediction error during the kth trial. The parameters were $\eta = 0.25$, $a = 2$, and the initial value set as $\sigma_0 = 4$.

5.1 Task Decomposition in Space: Nonlinear Control. We first used two modules, each of which had a linear dynamic model (see equation 3.9) and a quadratic reward model (see equation 3.10). The centers of the local linear dynamic models were initially placed randomly with the angular component in $[-\pi, \pi]$.

Each trial was started from a random position of the pendulum and lasted for 30 seconds.

Figure 4 shows an example of swing-up performance from the bottom position. Initially, the first prediction model predicts the pendulum motion better than the second one, so the responsibility signal λ_1 becomes close to 1. Thus, the output of the first RL controller u_1, which destabilizes the bottom position, is used for control. As the pendulum is driven away from the bottom, the second prediction model predicts the movement better, so λ_2 becomes higher and the second RL controller takes over and stabilizes the upright position.

Figure 5 shows the changes of linear prediction models and quadratic reward models before and after learning. The two linear prediction models approximated the nonlinear gravity term. The first model predicted the negative feedback acceleration around the equilibrium state with the pendulum hanging down. The second model predicted the positive feedback acceleration around the unstable equilibrium with the pendulum raised up. The two reward models also approximated the cosine reward function using parabolic curves.

Figure 6 shows the dynamic and reward models when there were eight modules. Two modules were specialized for the bottom position, three modules were specialized near the top position, and two other modules were centered somewhere in between. The result shows that proper modularization is possible even when there are redundant modules.

Figure 7 compares the time course of learning by MLQC with two, four, and eight modules and a nonmodular actor-critic (Doya, 2000). Learning was fastest with two modules. The addition of redundant modules resulted in more variability in the time course of learning. This is because there were multiple possible ways of modular decomposition, and due to the variability of the sample trajectories, it took longer for modular decomposition to stabilize. Nevertheless, learning by the eight-module MLQC was still much faster than by the nonmodular architecture.

An interesting feature of the MLQC strategy is that qualitatively different controllers are derived by the solutions of the Riccati equations, 3.12. The controller at the bottom is a positive feedback controller that destabilizes the equilibrium where the reward is minimal, while the controller at the top is a typical linear quadratic regulator that stabilizes the upright state. Another important feature of the MLQC is that the modules were flexibly switched simply based on the prediction errors. Successful swing up was achieved without any top-down planning of the complex sequence.

5.2 Task Decomposition in Time: Nonstationary Pendulum. We then tested the effectiveness of the MMRL architecture for the nonlinear and nonstationary control tasks in which mass m and length l of the pendulum were changed every trial.

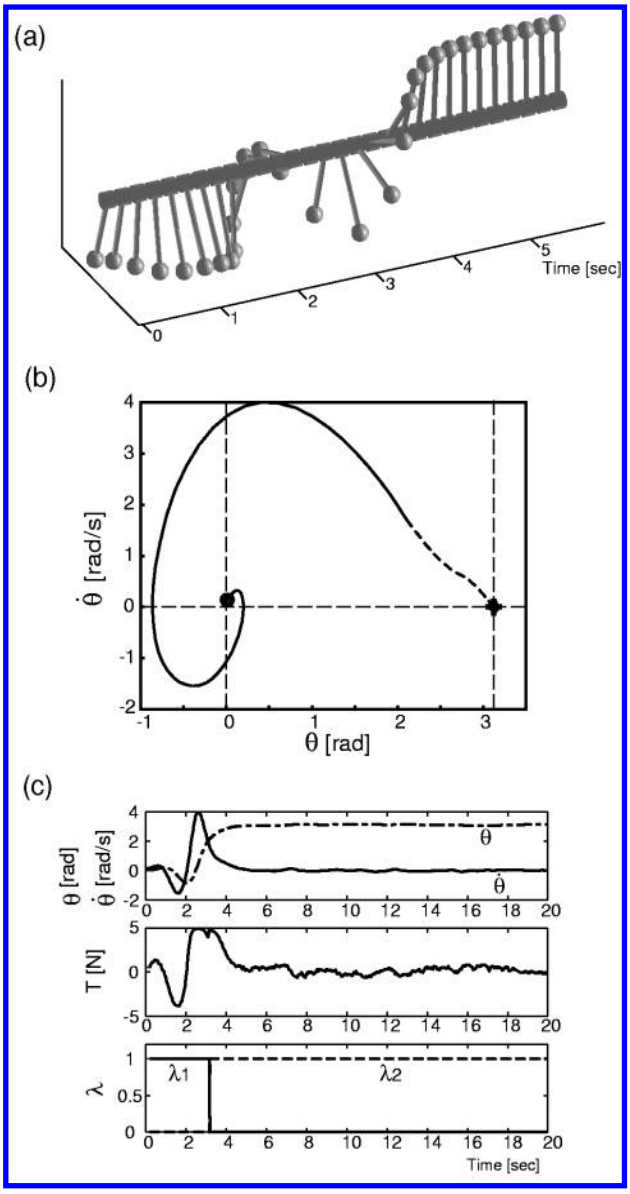

Figure 4: (a) Example of swing-up performance. Dynamics are given by $ml^2\ddot{\theta} = -mgl\sin\theta - \mu\dot{\theta} + T$. Physical parameters are $m = l = 1$, $g = 9.8$, $\mu = 0.1$, and $T^{\max} = 5.0$. (b) Trajectory from the initial state (0[rad],0.1[rad/s]). o: start, +: goal. Solid line: module 1. Dashed line: module 2. (c) Time course of the state (top), the action (middle), and the responsibility signal (bottom).

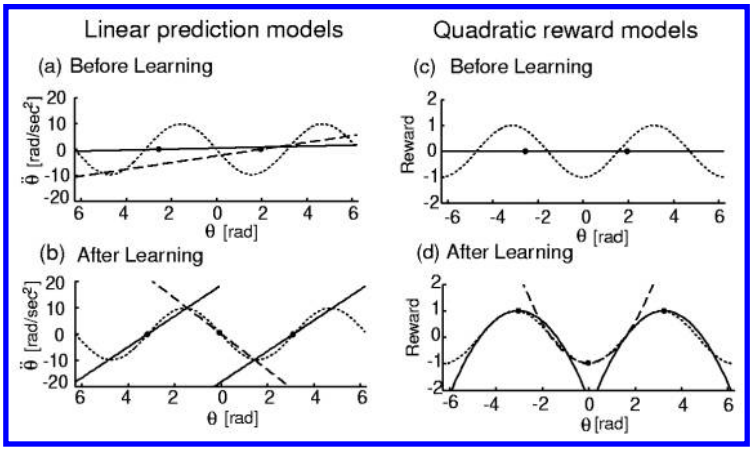

Figure 5: Development of state and reward prediction of models. (a,b) Outputs of state prediction models (a) before and (b) after learning. (c,d) Outputs of the reward prediction model (c) before and (d) after learning. Solid line: module 1. Dashed line: module 2; dotted line: targets ($\ddot{x}$ and r). ○: centers of spatial responsibility prediction $\mathbf{c}_i$.

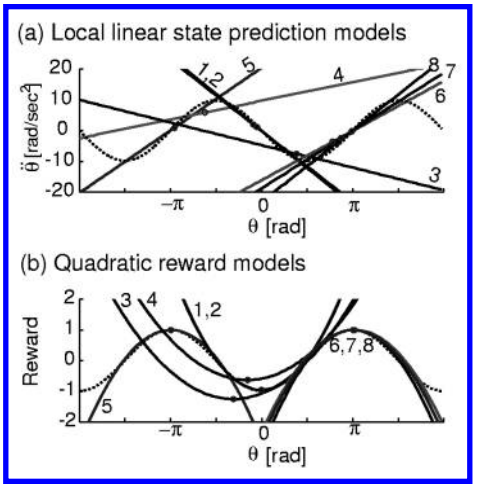

Figure 6: Outputs of eight modules. (a) State prediction models. (b) Reward models.

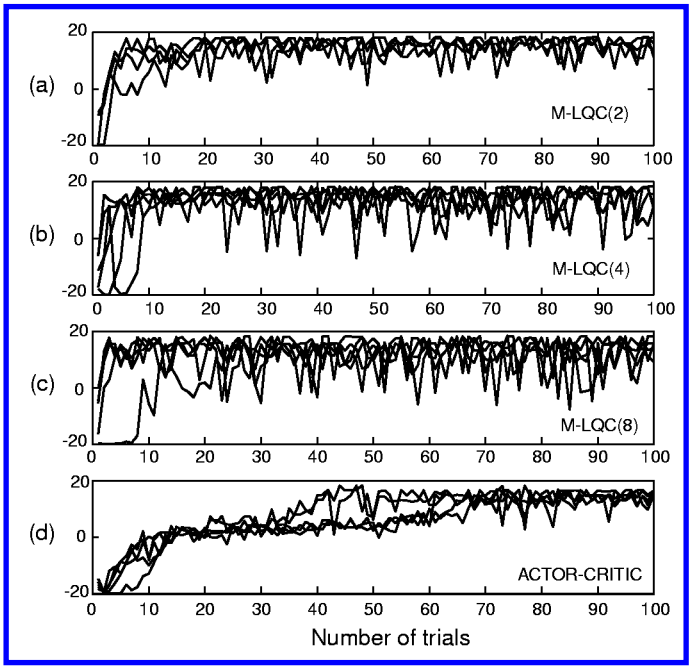

Figure 7: Learning curves for the pendulum swing-up task. The cumulative reward $\int_0^{20} r(t)\,dt$ during each trial is shown for five simulation runs. (a) Two modules. (b) Four modules. (c) Eight modules. (d) Nonmodular architecture.

We used four modules, each of which had a linear dynamic model (see equation 3.9) and a quadratic reward model (see equation 3.10). The centers x_i of the local linear prediction models were initially set randomly. Each trial was started from a random position with $\theta \in [-\pi/4, \pi/4]$ and lasted for 40 seconds.

We implemented responsibility prediction with $\tau_c = 50$, $\tau_M = 200$, and $\tau_p = 0.1$. The parameters of annealing were $\eta = 0.1$, $a = 2$, and an initial value of $\sigma_0 = 10$.

In the first 50 trials, the physical parameters were fixed at $\{m = 1.0, l = 1.0\}$. Figure 8a shows the change in the position gain ($\{A_{21}\} = \frac{\partial \ddot{\theta}}{\partial \theta}$) of the four prediction models. The control performance is shown in Figure 8b. Figures 8c, 8d, and 8e show the outputs of prediction models in the section of $\{\theta = 0, T = 0\}$.

Initial position gains are set randomly (see Figure 8c). After 50 trials, both modules 1 and 2 specialized in the bottom region ($\theta \simeq 0$) and learned similar prediction models. Modules 3 and 4 also learned the same prediction model in the top region ($\theta \simeq \pi$) (see Figure 8d). Accordingly, the RL controllers in modules 1 and 2 learned a reward model with a minimum near $(0, 0)'$, and

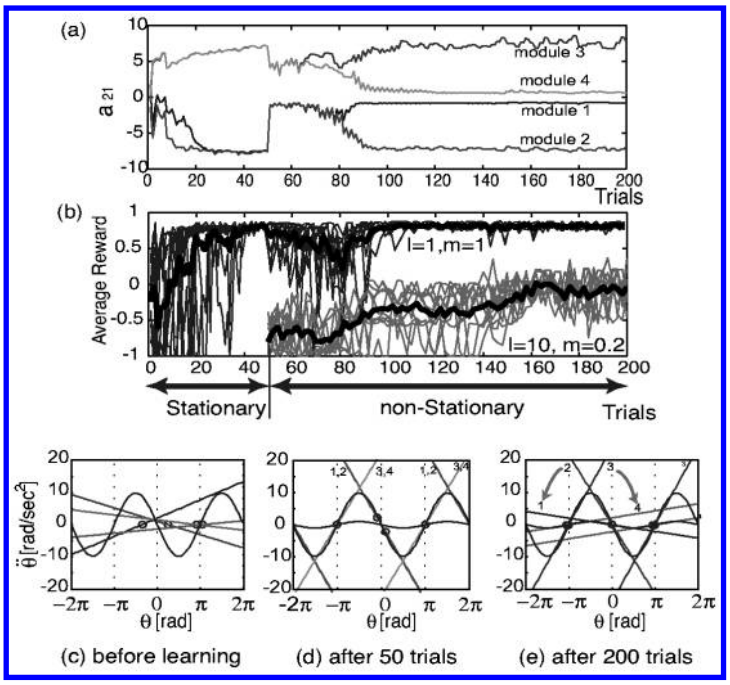

Figure 8: Time course of learning and changes of the prediction models. (a) Changes of a coefficient $A_{21} = \frac{\partial \ddot{\theta}}{\partial \theta}$ of the four prediction models, coefficient with angle. (b) Change of average reward during each trial. Thin lines: results of 10 simulation runs. Thick line: average to 10 simulation runs. Note that the average reward with the new, longer pendulum was lower even after successful learning because of its longer period of swinging. (c,d, and e) Linear prediction models in the section of $\{\theta = 0, T = 0\}$ (c) before learning, (d) after 50 trials with fixed parameters, and (e) after 150 trials with changing parameters. Slopes of linear models correspond to A_{21} shown in *a*.

a destabilizing feedback policy was given by equations 2.15 through 2.17. Modules 3 and 4 also learned a reward model with a peak near $(\pi, 0)'$ and implemented a stabilizing feedback controller.

In 50 to 200 trials, the parameters of the pendulum were switched between $\{m = 1, l = 1.0\}$ and $\{m = 0.2, l = 10.0\}$ in each trial. At first, the degenerated modules tried to follow the alternating environment (see Figure 8a), and thus swing up was not successful for the new, longer pendulum. The performance for the shorter pendulum was also disturbed (see Figure 8b). After about 80 trials, the prediction models gradually specialized in either new or learned dynamics (see Figure 8e), and successful swing up was achieved for both the shorter and longer pendulums.

We found similar module specialization in 6 of 10 simulation runs. In 4 other runs, due to the bias in initial module allocation, three modules were aggregated in one domain (top or bottom) and one model covered the other domain during the stationary condition. However, after 150 trials in the nonstationary condition, module specialization, as shown in Figure 8e, was achieved.

6 Discussion

We proposed an MMRL architecture that decomposes a nonlinear or nonstationary task in space and time based on the local predictability of the system dynamics. We tested the performance of the MMRL in both nonlinear and nonstationary control tasks. It was shown in simulations of the pendulum swing-up task that multiple prediction models were successfully trained and corresponding model-based controllers were derived.

The modules were specialized for different domains in the state space. It was also confirmed in a nonstationary pendulum swing-up task that available modules are flexibly allocated for different domains in space and time based on the task demands.

The modular control architecture using multiple prediction models was proposed by Wolpert and Kawato as a computational model of the cerebellum (Wolpert et al., 1998; Wolpert & Kawato, 1998). Imamizu et al. (1997, 2000) showed in fMRI experiments of novel tool use that a large area of the cerebellum is activated initially, and then a smaller area remains active after long training. They proposed that such local activation spots are the neural correlates of internal models of tools (Imamizu et al., 2000). They also suggested that internal models of different tools are represented in separated areas in the cerebellum (Imamizu et al., 1997). Our simulation results in a nonstationary environment can provide a computational account of these fMRI data. When a new task is introduced, many modules initially compete to learn it. However, after repetitive learning, only a subset of modules are specialized and recruited for the new task.

One might argue whether MLQC is a reinforcement learning architecture since it uses LQ controllers that were calculated off-line. However, when the linear dynamic models and quadratic reward models are learned on-line, as in our simulations, the entire system realizes reinforcement learning. One limitation of MLQC architecture is that the reward function should have helpful gradients in each modular domain. A method for backpropagating the value function of the successor module as the effective reward for the predecessor module is under development.

In order to construct a hierarchical RL system, it appears necessary to combine both top-down and bottom-up approaches for task decomposition. The MMRL architecture provides one solution for the bottom-up approach. Combination of this bottom-up mechanism with a top-down mechanism is the subject of our ongoing study.

Appendix: Modified Compositional Q-Learning

On each time step, the gating variable $g_i(t)$ is given by the prior probability of module selection, in this case from the assumption of temporal continuity, equation 2.22:

$$g_i(t) = \frac{\lambda_i(t-1)^\alpha}{\sum_{j=1}^n \lambda_j(t-1)^\alpha}. \tag{A.1}$$

The composite Q-values for state $x(t)$ are then computed by

$$\hat{Q}(x(t), u) = \sum_{i=1}^n g_i(t) Q_i(x(t), u), \tag{A.2}$$

and an action $u(t)$ is selected by

$$P(u \mid x(t)) = \frac{e^\beta \hat{Q}(x(t), u)}{\sum_{v \in U} e^\beta \hat{Q}(x(t), v)}. \tag{A.3}$$

After the reward $r(x(t), u(t))$ is acquired and the state changes to $x(t+1)$, the TD error for the module i is given by

$$e_i(t) = r(x(t), u(t)) + \gamma \max_u \hat{Q}(x(t+1), u) - Q_i(x(t), u(t)). \tag{A.4}$$

From gaussian assumption of value prediction error, the likelihood of module i is given by

$$P(e_i(t) \mid i) = e^{-\frac{1}{2\sigma^2} e_i(t)^2}, \tag{A.5}$$

and thus the responsibility signal, or the posterior probability for selecting module i, is given by

$$\lambda_i(t) = \frac{g_i(t) e^{-\frac{1}{2\sigma^2} e_i(t)^2}}{\sum_j g_j(t) e^{-\frac{1}{2\sigma^2} e_j(t)^2}}. \tag{A.6}$$

The Q values of each module are updated with the weighted TD error $\lambda_i(t) e_i(t)$ as the error signal.

The discount factor was set as $\gamma = 0.9$ and the greediness parameters as $\beta = 1$ for both MMRL and CQ-L. The decay parameter of temporal responsibility predictor was $\alpha = 0.8$ for MMRL. We tried different values of α for CQ-L without success. The value used in Figures 2 and 3 was $\alpha = 0.99$.

Acknowledgments

We thank Masahiko Haruno, Daniel Wolpert, Chris Atkeson, Jun Tani, Hidenori Kimura, and Raju Bapi for helpful discussions.

References

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 834–846.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific.

Cacciatore, T. W., & Nowlan, S. J. (1994). Mixture of controllers for jump linear and non-linear plants. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing system, 6*. San Mateo, CA: Morgan Kaufmann.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. In C. L. Giles, S. J. Hanson, & J. D. Cowan (Eds.), *Advances in neural information processing systems, 5* (pp. 271–278). San Mateo, CA: Morgan Kaufmann.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation, 12*, 215–245.

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. M., & Kawato, M. (1999). Multiple paired forward-inverse models for human motor learning and control. In M. S. Kearns, S. A. Solla, & D. A. Cohen (Eds.), *Advances in neural information processing systems, 11* (pp. 31–37). Cambridge, MA: MIT Press.

Haruno, M., Wolpert, D. M., & Kawato, M. (2001). MOSAIC model for sensorimotor learning and control. *Neural Computation, 13*, 2201–2220.

Imamizu, H., Miyauchi, S., Sasaki, Y., Takino, R., Pütz, B., & Kawato, M. (1997). Separated modules for visuomotor control and learning in the cerebellum: A functional MRI study. In A. W. Toga, R. S. J. Frackowiak, & J. C. Mazziotta (Eds.), *NeuroImage: Third International Conference on Functional Mapping of the Human Brain* (Vol. 5). Copenhagen, Denmark: Academic Press.

Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Pütz, B., Yoshioka, T., & Kawato, M. (2000). Human cerebellar activity reflecting an acquired internal model of a new tool. *Nature, 403*, 192–195.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*, 79–87.

Littman, M., Cassandra, A., & Kaelbling, L. (1995). Learning policies for partially observable environments: Scaling up. In A. Prieditis & S. Russel (Eds.), *Machine Learning: Proceedings of the 12th International Conference* (pp. 362–370). San Mateo, CA: Morgan Kaufmann.

Morimoto, J. & Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems, 36*, 37–51.

Narendra, K. S., Balakrishnan, J., & Ciliz, M. K. (1995, June). Adaptation and learning using multiple models, switching and tuning. *IEEE Control Systems Magazine*, 37–51.

Parr, R., & Russel, S. (1998). Reinforcement learning with hierarchies of machines. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), *Advances in neural information processing systems, 10* (pp. 1043–1049). Cambridge, MA: MIT Press.

Pawelzik, K., Kohlmorge, J., & Müller, K. R. (1996). Annealed competition of experts for a segmentation and classification of switching dynamics. *Neural Computation, 8*, 340–356.

Schaal, S., & Atkeson, C. G. (1996). From isolation to cooperation: An alternative view of a system of experts. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems, 8* (pp. 605–611). Cambridge, MA: MIT Press.

Singh, S. P. (1992). Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning, 8*, 323–340.

Sutton, R. S. (1988). Learning to predict by the methods of temporal difference. *Machine Learning, 3*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.

Sutton, R., Precup, D., & Singh, S. (1999). Between MDPS and semi-MDPS: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence, 112*, 181–211.

Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems. *Neural Networks, 12*, 1131–1141.

Wiering, M., & Schmidhuber, J. (1998). HQ-learning. *Adaptive Behavior, 6*, 219–246.

Wolpert, D. M., & Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience, 3*, 1212–1217.

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

Wolpert, D. M., Miall, R. C., & Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Sciences, 2*, 338–347.

Received June 12, 2000; accepted November 8, 2001.

This article has been cited by:

1. Jiexin Wang, Stefan Elfwing, Eiji Uchibe. 2021. Modular deep reinforcement learning from reward and punishment for robot navigation. *Neural Networks* **135**, 115-126. [Crossref]
2. Seng Bum Michael Yoo, Benjamin Yost Hayden, John M. Pearson. 2021. Continuous decisions. *Philosophical Transactions of the Royal Society B: Biological Sciences* **376**:1819, 20190664. [Crossref]
3. Y. Vandaele, S. H. Ahmed. 2021. Habit, choice, and addiction. *Neuropsychopharmacology* **46**:4, 689-698. [Crossref]
4. Jeffery Dick, Pawel Ladosz, Eseoghene Ben-Iwhiwhu, Hideyasu Shimadzu, Peter Kinnell, Praveen K. Pilly, Soheil Kolouri, Andrea Soltoggio. 2020. Detecting Changes and Avoiding Catastrophic Forgetting in Dynamic Partially Observable Environments. *Frontiers in Neurorobotics* **14**. . [Crossref]
5. Wei Liu, Evan J. Livesey, Harald Lachnit, Hilary J. Don, Anna Thorwart. 2020. Does learning history shape the associability of outcomes? Further tests of the outcome predictability effect. *PLOS ONE* **15**:12, e0243434. [Crossref]
6. Sean Gillen, Marco Molnar, Katie Byl. Combining Deep Reinforcement Learning And Local Control For The Acrobot Swing-up And Balance Task 4129-4134. [Crossref]
7. Jun Tani, Jeffrey White. 2020. Cognitive neurorobotics and self in the shared world, a focused review of ongoing research. *Adaptive Behavior* **13**, 105971232096215. [Crossref]
8. Junya Morita, Kazuhisa Miwa, Akihiro Maehigashi, Hitoshi Terai, Kazuaki Kojima, Frank E. Ritter. 2020. Cognitive Modeling of Automation Adaptation in a Time Critical Task. *Frontiers in Psychology* **11**. . [Crossref]
9. Yazhou Hu, Wenxue Wang, Hao Liu, Lianqing Liu. 2020. Reinforcement Learning Tracking Control for Robotic Manipulator With Kernel-Based Dynamic Model. *IEEE Transactions on Neural Networks and Learning Systems* **31**:9, 3570-3578. [Crossref]
10. Amandeep Singh Bhatia, Mandeep Kaur Saggi, Amit Sundas, Jatinder Ashta. Reinforcement Learning 281-303. [Crossref]
11. Mitsuo Kawato, Shogo Ohmae, Huu Hoang, Terry Sanger. 2020. 50 Years Since the Marr, Ito, and Albus Models of the Cerebellum. *Neuroscience* . [Crossref]
12. Momchil S. Tomov, Samyukta Yagati, Agni Kumar, Wanqian Yang, Samuel J. Gershman. 2020. Discovery of hierarchical representations for efficient planning. *PLOS Computational Biology* **16**:4, e1007594. [Crossref]
13. Siddhant Gangapurwala, Alexander Mitchell, Ioannis Havoutis. 2020. Guided Constrained Policy Optimization for Dynamic Quadrupedal Robot Locomotion. *IEEE Robotics and Automation Letters* **5**:2, 3642-3649. [Crossref]
14. Randall C. O'Reilly, Ananta Nair, Jacob L. Russin, Seth A. Herd. 2020. How Sequential Interactive Processing Within Frontostriatal Loops Supports a Continuum of Habitual to Controlled Processing. *Frontiers in Psychology* **11**. . [Crossref]
15. Tao Bian, Daniel M. Wolpert, Zhong-Ping Jiang. 2020. Model-Free Robust Optimal Feedback Mechanisms of Biological Motor Control. *Neural Computation* **32**:3, 562-595. [Abstract] [Full Text] [PDF] [PDF Plus]
16. Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, Mengling Feng. 2020. Reinforcement Learning for Clinical Decision Support in Critical Care: Comprehensive Review. *Journal of Medical Internet Research* **22**:7, e18477. [Crossref]
17. Parvin Malekzadeh, Mohammad Salimibeni, Arash Mohammadi, Akbar Assa, Konstantinos N. Plataniotis. 2020. MM-KTD: Multiple Model Kalman Temporal Differences for Reinforcement Learning. *IEEE Access* **8**, 128716-128729. [Crossref]
18. Song Chen, Junpeng Jiang, Xiaofang Zhang, Jinjin Wu, Gongzheng Lu. GAN-Based Planning Model in Deep Reinforcement Learning 323-334. [Crossref]
19. Stephanie M. Groman. The Neurobiology of Impulsive Decision-Making and Reinforcement Learning in Nonhuman Animals 23-52. [Crossref]
20. Dongjae Kim, Geon Yeong Park, John P. O'Doherty, Sang Wan Lee. 2019. Task complexity interacts with state-space uncertainty in the arbitration between model-based and model-free learning. *Nature Communications* **10**:1. . [Crossref]
21. Daria Yakovleva, Artem Popov, Andrey Filchenkov. Real-Time Bidding with Soft Actor-Critic Reinforcement Learning in Display Advertising 373-382. [Crossref]
22. Yazhou Hu, Wenxue Wang, Hao Liu, Lianqing Liu. Robotic Tracking Control with Kernel Trick-based Reinforcement Learning 997-1002. [Crossref]
23. Domingo Esteban, Leonel Rozo, Darwin G. Caldwell. Hierarchical Reinforcement Learning for Concurrent Discovery of Compound and Composable Policies 1818-1825. [Crossref]
24. Yi Wu, Yuxin Wu, Aviv Tamar, Stuart Russell, Georgia Gkioxari, Yuandong Tian. Bayesian Relational Memory for Semantic Visual Navigation 2769-2779. [Crossref]

25. Peng Liu, Yingnan Zhao, Wei Zhao, Xianglong Tang, Zichan Yang. 2019. An exploratory rollout policy for imagination-augmented agents. *Applied Intelligence* **49**:10, 3749-3764. [Crossref]
26. Gaddi Blumrosen. Enhancing Healthcare Quality with Reinforcement Learning Modeling 1-4. [Crossref]
27. Meng Zhang, Ming-Gang Gan, Jie Chen, Zhong-Ping Jiang. 2019. Data-driven adaptive optimal control of linear uncertain systems with unknown jumping dynamics. *Journal of the Franklin Institute* **356**:12, 6087-6105. [Crossref]
28. Juan Pablo Mendoza, Reid Simmons, Manuela Veloso. 2019. Detection and correction of subtle context-dependent robot model inaccuracies using parametric regions. *The International Journal of Robotics Research* **38**:8, 887-909. [Crossref]
29. 2019. A Reinforcement Learning Architecture That Transfers Knowledge Between Skills When Solving Multiple Tasks. *IEEE Transactions on Cognitive and Developmental Systems* **11**:2, 292-317. [Crossref]
30. Maryam Hashemzadeh, Reshad Hosseini, Majid Nili Ahmadabadi. 2019. Exploiting Generalization in the Subspaces for Faster Model-Based Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* **30**:6, 1635-1650. [Crossref]
31. Maël Lebreton, Karin Bacily, Stefano Palminteri, Jan B. Engelmann. 2019. Contextual influence on confidence judgments in human reinforcement learning. *PLOS Computational Biology* **15**:4, e1006973. [Crossref]
32. Kaileigh A. Byrne, A. Ross Otto, Bo Pang, Christopher J. Patrick, Darrell A. Worthy. 2019. Substance use is associated with reduced devaluation sensitivity. *Cognitive, Affective, & Behavioral Neuroscience* **19**:1, 40-55. [Crossref]
33. Ruizhuo Song, Qinglai Wei, Qing Li. Multiple Actor-Critic Optimal Control via ADP 63-93. [Crossref]
34. Youna Vandaele, Patricia H. Janak. 2018. Defining the place of habit in substance use disorders. *Progress in Neuro-Psychopharmacology and Biological Psychiatry* **87**, 22-32. [Crossref]
35. Zejian Zhou, Hao Xu. Switching Deep Reinforcement Learning based Intelligent Online Decision Making for Autonomous Systems under Uncertain Environment 1453-1460. [Crossref]
36. Eiji Uchibe. 2018. Cooperative and Competitive Reinforcement and Imitation Learning for a Mixture of Heterogeneous Learning Modules. *Frontiers in Neurorobotics* **12**. . [Crossref]
37. Elsa Fouragnan, Chris Retzler, Marios G. Philiastides. 2018. Separate neural representations of prediction error valence and surprise: Evidence from an fMRI meta-analysis. *Human Brain Mapping* **39**:7, 2887-2906. [Crossref]
38. Zachary Renwick, Dawn Tilbury, Ella Atkins. A Reinforcement Learning Based Adaptive Supervisor for Multiple Model Adaptive Estimation and Control 0216-0221. [Crossref]
39. Dorothea Koert, Guilherme Maeda, Gerhard Neumann, Jan Pcters. Learning Coupled Forward-Inverse Models with Combined Prediction Errors 2433-2439. [Crossref]
40. Nicholas T. Franklin, Michael J. Frank. 2018. Compositional clustering in task structure learning. *PLOS Computational Biology* **14**:4, e1006116. [Crossref]
41. Laurent Dollé, Ricardo Chavarriaga, Agnès Guillot, Mehdi Khamassi. 2018. Interactions of spatial strategies producing generalization gradient and blocking: A computational approach. *PLOS Computational Biology* **14**:4, e1006092. [Crossref]
42. Wolfgang M Pauli, Jeffrey Cockburn, Eva R Pool, Omar D Pérez, John P O'Doherty. 2018. Computational approaches to habits in a model-free world. *Current Opinion in Behavioral Sciences* **20**, 104-109. [Crossref]
43. Angela J Langdon, Melissa J Sharpe, Geoffrey Schoenbaum, Yael Niv. 2018. Model-based predictions for dopamine. *Current Opinion in Neurobiology* **49**, 1-7. [Crossref]
44. Anne G. E. Collins, Michael J. Frank. 2018. Within- and across-trial dynamics of human EEG reveal cooperative interplay between reinforcement learning and working memory. *Proceedings of the National Academy of Sciences* **115**:10, 2502-2507. [Crossref]
45. Deanna M. Barch, Adam Culbreth, Julia Sheffield. Systems Level Modeling of Cognitive Control in Psychiatric Disorders 145-173. [Crossref]
46. Mimi Liljeholm. Instrumental Divergence and Goal-Directed Choice 27-48. [Crossref]
47. Ibrahim Ahmed, Hamed Khorasgani, Gautam Biswas. 2018. Comparison of Model Predictive and Reinforcement Learning Methods for Fault Tolerant Control. *IFAC-PapersOnLine* **51**:24, 233-240. [Crossref]
48. Jan Skach, Bahare Kiumarsi, Frank L. Lewis, Ondrej Straka. 2018. Actor-Critic Off-Policy Learning for Optimal Control of Multiple-Model Discrete-Time Systems. *IEEE Transactions on Cybernetics* **48**:1, 29-40. [Crossref]
49. Fan-Yun Sun, Yen-Yu Chang, Yueh-Hua Wu, Shou-De Lin. Designing Non-greedy Reinforcement Learning Agents with Diminishing Reward Shaping 297-302. [Crossref]
50. Nicola Catenacci Volpi, Yan Wu, Dimitri Ognibene. Towards event-based MCTS for autonomous cars 420-427. [Crossref]
51. Katia M. Harlé, Dalin Guo, Shunan Zhang, Martin P. Paulus, Angela J. Yu. 2017. Anhedonia and anxiety underlying depressive symptomatology have distinct effects on reward-based decision-making. *PLOS ONE* **12**:10, e0186473. [Crossref]

52. Yunduan Cui, Takamitsu Matsubara, Kenji Sugimoto. 2017. Kernel dynamic policy programming: Applicable reinforcement learning to robot systems with high dimensional states. *Neural Networks* **94**, 13-23. [Crossref]
53. Cristóbal Moënne-Loccoz, Rodrigo C. Vergara, Vladimir López, Domingo Mery, Diego Cosmelli. 2017. Modeling Search Behaviors during the Acquisition of Expertise in a Sequential Decision-Making Task. *Frontiers in Computational Neuroscience* **11**. . [Crossref]
54. George I. Christopoulos, Xiao-Xiao Liu, Ying-yi Hong. 2017. Toward an Understanding of Dynamic Moral Decision Making: Model-Free and Model-Based Learning. *Journal of Business Ethics* **144**:4, 699-715. [Crossref]
55. Trung Thanh Nguyen, Tomi Silander, Zhuoru Li, Tze-Yun Leong. 2017. Scalable transfer learning in heterogeneous, dynamic environments. *Artificial Intelligence* **247**, 70-94. [Crossref]
56. Justin Reber, Justin S. Feinstein, John P. O'Doherty, Mimi Liljeholm, Ralph Adolphs, Daniel Tranel. 2017. Selective impairment of goal-directed decision-making following lesions to the human ventromedial prefrontal cortex. *Brain* **140**:6, 1743-1756. [Crossref]
57. Greg Foderaro, Ashleigh Swingler, Silvia Ferrari. 2017. A Model-Based Approach to Optimizing <italic>Ms. Pac-Man</italic> Game Strategies in Real Time. *IEEE Transactions on Computational Intelligence and AI in Games* **9**:2, 153-165. [Crossref]
58. Andrei Marinescu, Ivana Dusparic, Siobhán Clarke. 2017. Prediction-Based Multi-Agent Reinforcement Learning in Inherently Non-Stationary Environments. *ACM Transactions on Autonomous and Adaptive Systems* **12**:2, 1-23. [Crossref]
59. James Kozloski. Synaptic integrators implement inhibitory plasticity, eliminate loops and create a "winnerless" Network 1-4. [Crossref]
60. Taposh Banerjee, Miao Liu, Jonathan P. How. Quickest change detection approach to optimal control in Markov decision processes with model changes 399-405. [Crossref]
61. James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* **114**:13, 3521-3526. [Crossref]
62. Ruizhuo Song, Frank L. Lewis, Qinglai Wei. 2017. Off-Policy Integral Reinforcement Learning Method to Solve Nonlinear Continuous-Time Multiplayer Nonzero-Sum Games. *IEEE Transactions on Neural Networks and Learning Systems* **28**:3, 704-713. [Crossref]
63. Prachi Mistry, Mimi Liljeholm. 2016. Instrumental Divergence and the Value of Control. *Scientific Reports* **6**:1. . [Crossref]
64. Kumpati S. Narendra, Yu Wang, Snehasis Mukhopadhay. Fast Reinforcement Learning using multiple models 7183-7188. [Crossref]
65. Tejas Savalia, Anuj Shukla, Raju S. Bapi. 2016. A Unified Theoretical Framework for Cognitive Sequencing. *Frontiers in Psychology* **7**. . [Crossref]
66. Christian Jarvers, Tobias Brosch, André Brechmann, Marie L. Woldeit, Andreas L. Schulz, Frank W. Ohl, Marcel Lommerzheim, Heiko Neumann. 2016. Reversal Learning in Humans and Gerbils: Dynamic Control Network Facilitates Learning. *Frontiers in Neuroscience* **10**. . [Crossref]
67. Noor Shaker. Intrinsically motivated reinforcement learning: A promising framework for procedural content generation 1-8. [Crossref]
68. Xue Bin Peng, Glen Berseth, Michiel van de Panne. 2016. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics* **35**:4, 1-12. [Crossref]
69. Jun-Sheng Wang, Guang-Hong Yang. 2016. Data-driven output-feedback fault-tolerant control for unknown dynamic systems with faults changing system dynamics. *Journal of Process Control* **43**, 10-23. [Crossref]
70. Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, Hideki Asoh. 2016. Symbol emergence in robotics: a survey. *Advanced Robotics* **30**:11-12, 706-728. [Crossref]
71. Andreas L. Schulz, Marie L. Woldeit, Ana I. Gonçalves, Katja Saldeitis, Frank W. Ohl. 2016. Selective Increase of Auditory Cortico-Striatal Coherence during Auditory-Cued Go/NoGo Discrimination Learning. *Frontiers in Behavioral Neuroscience* **9**. . [Crossref]
72. Hugh Rabagliati, Chiara Gambi, Martin J. Pickering. 2016. Learning to predict or predicting to learn?. *Language, Cognition and Neuroscience* **31**:1, 94-105. [Crossref]
73. Frédéric Alexandre, Maxime Carrere. Modeling Neuromodulation as a Framework to Integrate Uncertainty in General Cognitive Architectures 324-333. [Crossref]

74. Seiji Ishihara, Harukazu Igarashi. 2016. Policy Gradient Reinforcement Learning with Separated Knowledge: Environmental Dynamics and Action-Values in Policies. *IEEJ Transactions on Electronics, Information and Systems* **136**:3, 282-289. [Crossref]
75. James Bonaiuto, Michael A. Arbib. 2015. Learning to grasp and extract affordances: the Integrated Learning of Grasps and Affordances (ILGA) model. *Biological Cybernetics* **109**:6, 639-669. [Crossref]
76. Jessica A. Cooper, Darrell A. Worthy, W. Todd Maddox. 2015. Chronic motivational state interacts with task reward structure in dynamic decision-making. *Cognitive Psychology* **83**, 40-53. [Crossref]
77. Matteo Colombo. 2015. For a Few Neurons More: Tractability and Neurally Informed Economic Modelling. *The British Journal for the Philosophy of Science* **66**:4, 713-736. [Crossref]
78. Kate M. Wassum, Alicia Izquierdo. 2015. The basolateral amygdala in reward learning and addiction. *Neuroscience & Biobehavioral Reviews* **57**, 271-283. [Crossref]
79. Bo Ryu, Nadeesha Ranasinghe, Wei-Min Shen, Kurt Turck, Michael Muccio. BioAIM: Bio-inspired Autonomous Infrastructure Monitoring 780-785. [Crossref]
80. Daniel Philip Venmani, Arai Kaoru. On Improving Clock Synchronization Accuracy for LTE-A Networks 1-5. [Crossref]
81. Wolfram Schultz. 2015. Neuronal Reward and Decision Signals: From Theories to Data. *Physiological Reviews* **95**:3, 853-951. [Crossref]
82. Bahare Kiumarsi, Frank L. Lewis, Daniel S. Levine. 2015. Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure. *Neurocomputing* **156**, 157-165. [Crossref]
83. Mimi Liljeholm, Simon Dunne, John P. O'Doherty. 2015. Differentiating neural systems mediating the acquisition vs. expression of goal-directed and habitual behavioral control. *European Journal of Neuroscience* **41**:10, 1358-1371. [Crossref]
84. Ruizhuo Song, Frank Lewis, Qinglai Wei, Hua-Guang Zhang, Zhong-Ping Jiang, Dan Levine. 2015. Multiple Actor-Critic Structures for Continuous-Time Optimal Control Using Input-Output Data. *IEEE Transactions on Neural Networks and Learning Systems* **26**:4, 851-865. [Crossref]
85. Xavier Lagorce, Sio-Hoi Ieng, Xavier Clady, Michael Pfeiffer, Ryad B. Benosman. 2015. Spatiotemporal features for asynchronous event-based data. *Frontiers in Neuroscience* **9**. . [Crossref]
86. Fumino Fujiyama, Susumu Takahashi, Fuyuki Karube. 2015. Morphological elucidation of basal ganglia circuits contributing reward prediction. *Frontiers in Neuroscience* **9**. . [Crossref]
87. John P O'Doherty, Sang Wan Lee, Daniel McNamee. 2015. The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences* **1**, 94-100. [Crossref]
88. Fatemeh Yavari. 2015. Does our brain use the same policy for interacting with people and manipulating different objects?. *Frontiers in Computational Neuroscience* **8**. . [Crossref]
89. J.P. O'Doherty. Neuroimaging Studies of Reinforcement-Learning 375-380. [Crossref]
90. Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* **61**, 85-117. [Crossref]
91. Yuta Tsuge, Tanagorn Jennawasin, Tatsuo Narikiyo, Michihiro Kawanishi. 2015. Nonlinear Control of Partially Known Systems Based on Polynomial Representation and Reinforcement Learning. *IEEJ Transactions on Electronics, Information and Systems* **135**:2, 215-224. [Crossref]
92. Wiem Zemzem, Moncef Tagina. 2015. A Novel Exploration/Exploitation Policy Accelerating Learning in Both Stationary and Non-Stationary Environment Navigation Tasks. *International Journal of Computer and Electrical Engineering* **7**:3, 149-158. [Crossref]
93. Tetsunari Inamura, Yoshihiko Nakamura. Stochastic Information Processing that Unifies Recognition and Generation of Motion Patterns: Toward Symbolical Understanding of the Continuous World 79-102. [Crossref]
94. Daniel L Elliott, Charles Anderson. Using supervised training signals of observable state dynamics to speed-up and improve reinforcement learning 1-8. [Crossref]
95. Etienne Koechlin. 2014. An evolutionary computational theory of prefrontal executive function in decision-making. *Philosophical Transactions of the Royal Society B: Biological Sciences* **369**:1655, 20130474. [Crossref]
96. Florian Lesaint, Olivier Sigaud, Mehdi Khamassi. 2014. Accounting for Negative Automaintenance in Pigeons: A Dual Learning Systems Approach and Factored Representations. *PLoS ONE* **9**:10, e111050. [Crossref]
97. M. Donoso, A. G. E. Collins, E. Koechlin. 2014. Foundations of human reasoning in the prefrontal cortex. *Science* **344**:6191, 1481-1486. [Crossref]
98. Eiji Uchibe, Kenji Doya. Combining learned controllers to achieve new goals based on linearly solvable MDPs 5252-5259. [Crossref]

99. Peter Dayan. 2014. Rationalizable Irrationalities of Choice. *Topics in Cognitive Science* **6**:2, 204-228. [Crossref]
100. Florian Lesaint, Olivier Sigaud, Shelly B. Flagel, Terry E. Robinson, Mehdi Khamassi. 2014. Modelling Individual Differences in the Form of Pavlovian Conditioned Approach Responses: A Dual Learning Systems Approach with Factored Representations. *PLoS Computational Biology* **10**:2, e1003466. [Crossref]
101. Daisuke Uragami, Tatsuji Takahashi, Yoshiki Matsuo. 2014. Cognitively inspired reinforcement learning architecture and its application to giant-swing motion control. *Biosystems* **116**, 1-9. [Crossref]
102. Sang Wan Lee, Shinsuke Shimojo, John P. O'Doherty. 2014. Neural Computations Underlying Arbitration between Model-Based and Model-free Learning. *Neuron* **81**:3, 687-699. [Crossref]
103. Emmanuel Hadoux, Aurélie Beynier, Paul Weng. Solving Hidden-Semi-Markov-Mode Markov Decision Problems 176-189. [Crossref]
104. Vieri G. Santucci, Gianluca Baldassarre, Marco Mirolli. Cumulative Learning Through Intrinsic Reinforcements 107-122. [Crossref]
105. Anitha Vijaya Kumar, Akilandeswari Jeyapal. 2014. Self-Adaptive Trust Based ABR Protocol for MANETs Using Q -Learning. *The Scientific World Journal* **2014**, 1-9. [Crossref]
106. Romain D. Cazé, Matthijs A. A. van der Meer. 2013. Adaptive properties of differential learning rates for positive and negative outcomes. *Biological Cybernetics* **107**:6, 711-719. [Crossref]
107. Ignasi Cos, Mehdi Khamassi, Benoıˆt Girard. 2013. Modelling the learning of biomechanics and visual planning for decision-making of motor actions. *Journal of Physiology-Paris* **107**:5, 399-408. [Crossref]
108. Anna Lisa Ciancio, Loredana Zollo, Gianluca Baldassarre, Daniele Caligiore, Eugenio Guglielmelli. 2013. The Role of Learning and Kinematic Features in Dexterous Manipulation: A Comparative Study with Two Robotic Hands. *International Journal of Advanced Robotic Systems* **10**:10, 340. [Crossref]
109. Fatemeh Yavari, Farzad Towhidkhah, Mohammad Ali Ahmadi-Pajouh. 2013. Are fast/slow process in motor adaptation and forward/inverse internal model two sides of the same coin?. *Medical Hypotheses* **81**:4, 592-600. [Crossref]
110. Ray J. Dolan, Peter Dayan. 2013. Goals and Habits in the Brain. *Neuron* **80**:2, 312-325. [Crossref]
111. Jun-ichi Iwata, Keisetsu Shima, Jun Tanji, Hajime Mushiake. 2013. Neurons in the cingulate motor area signal context-based and outcome-based volitional selection of action. *Experimental Brain Research* **229**:3, 407-417. [Crossref]
112. Jane Garrison, Burak Erdeniz, John Done. 2013. Prediction error in reinforcement learning: A meta-analysis of neuroimaging studies. *Neuroscience & Biobehavioral Reviews* **37**:7, 1297-1310. [Crossref]
113. Akihiko Yamaguchi, Jun Takamatsu, Tsukasa Ogasawara. 2013. DCOB: Action space for reinforcement learning of high DoF robots. *Autonomous Robots* **34**:4, 327-346. [Crossref]
114. Wolfram Schultz. 2013. Updating dopamine reward signals. *Current Opinion in Neurobiology* **23**:2, 229-238. [Crossref]
115. Yuka Ariki, Sang-Ho Hyon, Jun Morimoto. 2013. Extraction of primitive representation from captured human movements and measured ground reaction force to generate physically consistent imitated behaviors. *Neural Networks* **40**, 32-43. [Crossref]
116. Charlotte Prévost, Daniel McNamee, Ryan K. Jessup, Peter Bossaerts, John P. O'Doherty. 2013. Evidence for Model-based Computations in the Human Amygdala during Pavlovian Conditioning. *PLoS Computational Biology* **9**:2, e1002918. [Crossref]
117. Constantin A. Rothkopf, Dana H. Ballard. Learning and Coordinating Repertoires of Behaviors with Common Reward: Credit Assignment and Module Activation 99-125. [Crossref]
118. Luca Lonini, Christos Dimitrakakis, Constantin Rothkopf, Jochen Triesch. Generalization and Interference in Human Motor Control 155-176. [Crossref]
119. Andrew S. Kayser, Mark D'Esposito. 2013. Abstract Rule Learning: The Differential Effects of Lesions in Frontal Cortex. *Cerebral Cortex* **23**:1, 230-240. [Crossref]
120. Marco Mirolli, Gianluca Baldassarre. Functions and Mechanisms of Intrinsic Motivations 49-72. [Crossref]
121. Peter Dayan. Exploration from Generalization Mediated by Multiple Controllers 73-91. [Crossref]
122. Gianluca Baldassarre, Marco Mirolli. Deciding Which Skill to Learn When: Temporal-Difference Competence-Based Intrinsic Motivation (TD-CB-IM) 257-278. [Crossref]
123. Arturo Ribes, Jesus Cerquides Bueno, Yiannis Demiris, Ramon Lopez de Mantaras. Context-GMM: Incremental learning of sparse priors for Gaussian mixture regression 1446-1451. [Crossref]
124. Norikazu Sugimoto, Jun Morimoto, Sang-Ho Hyon, Mitsuo Kawato. 2012. The eMOSAIC model for humanoid robot control. *Neural Networks* **29-30**, 8-19. [Crossref]

125. John P. O'Doherty. 2012. Beyond simple reinforcement learning: the computational neurobiology of reward-learning and valuation. *European Journal of Neuroscience* **35**:7, 987-990. [Crossref]
126. Anne Collins, Etienne Koechlin. 2012. Reasoning, Learning, and Creativity: Frontal Lobe Function and Human Decision-Making. *PLoS Biology* **10**:3, e1001293. [Crossref]
127. Norikazu Sugimoto, Masahiko Haruno, Kenji Doya, Mitsuo Kawato. 2012. MOSAIC for Multiple-Reward Environments. *Neural Computation* **24**:3, 577-606. [Abstract] [Full Text] [PDF] [PDF Plus]
128. Michael J. Frank, David Badre. 2012. Mechanisms of Hierarchical Reinforcement Learning in Corticostriatal Circuits 1: Computational Analysis. *Cerebral Cortex* **22**:3, 509-526. [Crossref]
129. Mimi Liljeholm, John P. O'Doherty. 2012. Anything You Can Do, You Can Do Better: Neural Substrates of Incentive-Based Performance Enhancement. *PLoS Biology* **10**:2, e1001272. [Crossref]
130. J. Vervaeke, T. P. Lillicrap, B. A. Richards. 2012. Relevance Realization and the Emerging Framework in Cognitive Science. *Journal of Logic and Computation* **22**:1, 79-99. [Crossref]
131. Johane Takeuchi, Osamu Shouno, Hiroshi Tsujino. 2012. A Flexible Behavioral Learning System with Modular Neural Networks. *Transactions of the Japanese Society for Artificial Intelligence* **27**:2, 92-102. [Crossref]
132. M. Mahdi Ghazaei Ardakani, Henrik Jorntell, Rolf Johansson. ORF-MOSAIC for adaptive control of a biomimetic arm 1273-1278. [Crossref]
133. William H Alexander, Joshua W Brown. 2011. Medial prefrontal cortex as an action-outcome predictor. *Nature Neuroscience* **14**:10, 1338-1344. [Crossref]
134. Norikazu Sugimoto, Jun Morimoto. Switching multiple LQG controllers based on bellman's optimality principle: Using full-state feedback to control a humanoid robot 3185-3191. [Crossref]
135. Chrisantha Fernando, Vera Vasas, Eörs Szathmáry, Phil Husbands. 2011. Evolvable Neuronal Paths: A Novel Basis for Information and Search in the Brain. *PLoS ONE* **6**:8, e23534. [Crossref]
136. Mark Ring, Tom Schaul, Juergen Schmidhuber. The two-dimensional organization of behavior 1-8. [Crossref]
137. Hazem Toutounji, Constantin A. Rothkopf, Jochen Triesch. Scalable reinforcement learning through hierarchical decompositions for weakly-coupled problems 1-7. [Crossref]
138. Anna Lisa Ciancio, Loredana Zollo, Eugenio Guglielmelli, Daniele Caligiore, Gianluca Baldassarre. Hierarchical reinforcement learning and central pattern generators for modeling the development of rhythmic manipulation skills 1-8. [Crossref]
139. Shingo Nakamura, Shuji Hashimoto. Application of hybrid learning strategy for manipulator robot 2465-2470. [Crossref]
140. Nathaniel D. Daw, Samuel J. Gershman, Ben Seymour, Peter Dayan, Raymond J. Dolan. 2011. Model-Based Influences on Humans' Choices and Striatal Prediction Errors. *Neuron* **69**:6, 1204-1215. [Crossref]
141. Hiroyuki Nakahara, Sivaramakrishnan Kaveri. 2010. Internal-Time Temporal Difference Model for Neural Value-Based Decision Making. *Neural Computation* **22**:12, 3062-3106. [Abstract] [Full Text] [PDF] [PDF Plus]
142. Chrisantha Fernando, Richard Goldstein, Eörs Szathmáry. 2010. The Neuronal Replicator Hypothesis. *Neural Computation* **22**:11, 2809-2857. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplemental Material]
143. Jürgen Schmidhuber. 2010. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* **2**:3, 230-247. [Crossref]
144. Ethan S. Bromberg-Martin, Masayuki Matsumoto, Simon Hong, Okihide Hikosaka. 2010. A Pallidus-Habenula-Dopamine Pathway Signals Inferred Stimulus Values. *Journal of Neurophysiology* **104**:2, 1068-1076. [Crossref]
145. R. D. Samson, M. J. Frank, Jean-Marc Fellous. 2010. Computational models of reinforcement learning: the role of dopamine as a reward signal. *Cognitive Neurodynamics* **4**:2, 91-105. [Crossref]
146. Jan Gläscher, Nathaniel Daw, Peter Dayan, John P. O'Doherty. 2010. States versus Rewards: Dissociable Neural Prediction Error Signals Underlying Model-Based and Model-Free Reinforcement Learning. *Neuron* **66**:4, 585-595. [Crossref]
147. J Zico Kolter, Christian Plagemann, David T Jackson, Andrew Y Ng, Sebastian Thrun. A probabilistic approach to mixed open-loop and closed-loop control, with application to extreme autonomous driving 839-845. [Crossref]
148. Norikazu Sugimoto, Jun Morimoto, Sang-Ho Hyon, Mitsuo Kawato. eMOSAIC Model for Humanoid Robot Control 447-457. [Crossref]
149. Mehran Emadi Andani, Fariba Bahrami, Parviz Jabehdar Maralani, Auke Jan Ijspeert. 2009. MODEM: a multi-agent hierarchical structure to model the human motor control system. *Biological Cybernetics* **101**:5-6, 361-377. [Crossref]
150. Masahiro Fujita. 2009. Intelligence Dynamics: a concept and preliminary experiments for open-ended learning agents. *Autonomous Agents and Multi-Agent Systems* **19**:3, 248-271. [Crossref]

151. Pinar Öztürk. 2009. Levels and Types of Action Selection: The Action Selection Soup. *Adaptive Behavior* **17**:6, 537-554. [Crossref]
152. Zeb Kurth-Nelson, A. David Redish. 2009. Temporal-Difference Reinforcement Learning with Distributed Representations. *PLoS ONE* **4**:10, e7362. [Crossref]
153. Aaron R. Seitz, Takeo Watanabe. 2009. The phenomenon of task-irrelevant perceptual learning. *Vision Research* **49**:21, 2604-2610. [Crossref]
154. Justin C. Sanchez, Jose C. Principe. Prerequesites for symbiotic brain-machine interfaces 1736-1741. [Crossref]
155. Luca Lonini, Laura Dipietro, Loredana Zollo, Eugenio Guglielmelli, Hermano Igo Krebs. 2009. An Internal Model for Acquisition and Retention of Motor Learning During Arm Reaching. *Neural Computation* **21**:7, 2009-2027. [Abstract] [Full Text] [PDF] [PDF Plus]
156. Michail Maniadakis, Panos Trahanias, Jun Tani. 2009. Explorations on artificial time perception. *Neural Networks* **22**:5-6, 509-517. [Crossref]
157. Ana L. C. Bazzan. 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems* **18**:3, 342-375. [Crossref]
158. Mehran Emadi Andani, Fariba Bahrami, Parviz Jabehdar Maralani. 2009. AMA-MOSAICI: An automatic module assigning hierarchical structure to control human motion based on movement decomposition. *Neurocomputing* **72**:10-12, 2310-2318. [Crossref]
159. Leszek Rybicki, Yuuya Sugita, Jun Tani. Reinforcement learning of multiple tasks using parametric bias 2732-2739. [Crossref]
160. Michail Maniadakis, Jun Tani, Panos Trahanias. Time perception in shaping cognitive neurodynamics of artificial agents 1993-2000. [Crossref]
161. Akihiko Yamaguchi, Jun Takamatsu, Tsukasa Ogasawara. Constructing action set from basis functions for reinforcement learning of robot control 2525-2532. [Crossref]
162. Justin C. Sanchez, Babak Mahmoudi, Jack DiGiovanna, Jose C. Principe. 2009. Exploiting co-adaptation for the design of symbiotic neuroprosthetic assistants. *Neural Networks* **22**:3, 305-315. [Crossref]
163. Hiroshi Tsujino, Johane Takeuchi, Osamu Shouno. Basal Ganglia Models for Autonomous Behavior Learning 328-350. [Crossref]
164. Camille Salaün, Vincent Padois, Olivier Sigaud. A Two-Level Model of Anticipation-Based Motor Learning for Whole Body Motion 229-246. [Crossref]
165. Johane Takeuchi, Osamu Shouno, Hiroshi Tsujino. Self-Referential Event Lists for Self-Organizing Modular Reinforcement Learning 228-235. [Crossref]
166. Masumi Ishikawa, Kosuke Ueno. Hierarchical Architecture with Modular Network SOM and Modular Reinforcement Learning 546-556. [Crossref]
167. Seiji Ishihara, Harukazu Igarashi. 2009. Behavior Learning Based on a Policy Gradient Method: Separation of Environmental Dynamics and State-Values in Policies. *IEEJ Transactions on Electronics, Information and Systems* **129**:9, 1737-1746. [Crossref]
168. Syuji KANEKO, Hakaru TAMUKOH, Kazuhiro TOKUNAGA, Tetsuo FURUKAWA. 2009. Hardware Implementation of Higher Rank of Self-Organizing Maps. *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics* **21**:5, 870-883. [Crossref]
169. Justin C. Sanchez, Renato Figueiredo, Jose Fortes, Jose C. Principe. Development of Symbiotic Brain-Machine Interfaces Using a Neurophysiology Cyberworkstation 606-615. [Crossref]
170. Makito Oku, Kazuyuki Aihara. 2008. Networked reinforcement learning. *Artificial Life and Robotics* **13**:1, 112-115. [Crossref]
171. Mehdi Khamassi, Antonius B. Mulder, Eiichi Tabuchi, Vincent Douchamps, Sidney I. Wiener. 2008. Anticipatory reward signals in ventral striatal neurons of behaving rats. *European Journal of Neuroscience* **28**:9, 1849-1866. [Crossref]
172. Shingo Nakamura, Shuji Hashimoto. 2008. Adaptive Modeling of Physical Systems Based on Affine Transform and its Application for Machine Learning. *Journal of Robotics and Mechatronics* **20**:5, 750-756. [Crossref]
173. N. Tomi, M. Gouko, K. Ito. Inaccuracy of internal models in force fields and complementary use of impedance control 393-398. [Crossref]
174. Eduardo C. Pinheiro. Multiple models oscillometric blood pressure monitor identification 319-322. [Crossref]
175. A. N. Hampton, P. Bossaerts, J. P. O'Doherty. 2008. Neural correlates of mentalizing-related computations during strategic interactions in humans. *Proceedings of the National Academy of Sciences* **105**:18, 6741-6746. [Crossref]
176. Jun Tani, Ryu Nishimoto, Jun Namikawa, Masato Ito. 2008. Codevelopmental Learning Between Human and Humanoid Robot Using a Dynamic Neural-Network Model. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**:1, 43-59. [Crossref]

177. Tadahiro Taniguchi, Tetsuo Sawaragi. 2008. Incremental acquisition of multiple nonlinear forward models based on differentiation process of schema model. *Neural Networks* **21**:1, 13-27. [Crossref]

178. Shingo Nakamura, Shuji Hashimoto. Hybrid learning strategy to solve pendulum swing-up problem for real hardware 1972-1977. [Crossref]

179. Hajime Fujita, Shin Ishii. 2007. Model-Based Reinforcement Learning for Partially Observable Games with Sampling-Based State Estimation. *Neural Computation* **19**:11, 3051-3087. [Abstract] [PDF] [PDF Plus]

180. Shingo Nakamura, Ryo Saegusa, Shuji Hashimoto. 2007. A Hybrid Learning Strategy for Real Hardware of Swing-Up Pendulum. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **11**:8, 972-978. [Crossref]

181. Ahad Harati, Majid Nili Ahmadabadi, Babak Nadjar Araabi. 2007. Knowledge-Based Multiagent Credit Assignment: A Study on Task Type and Critic Information. *IEEE Systems Journal* **1**:1, 55-67. [Crossref]

182. Mathieu Bertin, Nicolas Schweighofer, Kenji Doya. 2007. Multiple model-based reinforcement learning explains dopamine neuronal activity. *Neural Networks* **20**:6, 668-675. [Crossref]

183. M. Emadi Andani, F. Bahrami, P. Jabedar Maralani. A Biologically Inspired Modular Structure to Control the Sit-to-Stand Transfer of a Biped Robot 3016-3019. [Crossref]

184. Johane Takeuchi, Osamu Shouno, Hiroshi Tsujino. Modular Neural Networks for Reinforcement Learning with Temporal Intrinsic Rewards 1151-1156. [Crossref]

185. Stefan Schaal. 2007. The new robotics—towards human-centered machines. *HFSP Journal* **1**:2, 115-126. [Crossref]

186. Mitsuo Kawato, Kazuyuki Samejima. 2007. Efficient reinforcement learning: computational theories, neuroscience and robotics. *Current Opinion in Neurobiology* **17**:2, 205-212. [Crossref]

187. Pierre-Yves Oudeyer, Frdric Kaplan, Verena V. Hafner. 2007. Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Transactions on Evolutionary Computation* **11**:2, 265-286. [Crossref]

188. Shuhei Nishida, Kazuo Ishii, Tetsuo Furukawa. Self-Organizing Decision-Making System for AUV 506-511. [Crossref]

189. Hiroshi Imamizu, Satomi Higuchi, Akihiro Toda, Mitsuo Kawato. 2007. Reorganization of Brain Activity for Multiple Internal Models After Short But Intensive Training. *Cortex* **43**:3, 338-349. [Crossref]

190. Tadahiro Taniguchi, Tetsuo Sawaragi. 2007. Incremental acquisition of behaviors and signs based on a reinforcement learning schemata model and a spike timing-dependent plasticity network. *Advanced Robotics* **21**:10, 1177-1199. [Crossref]

191. Hideki Kadone, Yoshihiko Nakamura. Segmentation, Memorization, Recognition and Abstraction of Humanoid Motions Based on Correlations and Associative Memory 1-6. [Crossref]

192. Yu Ohigashi, Takashi Omori. 2006. Modeling of autonomous problem solving process by dynamic construction of task models in multiple tasks environment. *Neural Networks* **19**:8, 1169-1180. [Crossref]

193. Shogo Okada, Osamu Hasegawa. 2006. Incremental Learning, Recognition, and Generation of Time-Series Patterns Based on Self-Organizing Segmentation. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **10**:3, 395-408. [Crossref]

194. Shuhei Nishida, Kazuo Ishii, Tetsuo Furukawa. An Adaptive Neural Network Control System using mnSOM 1-6. [Crossref]

195. Akitoshi Ogawa, Takashi Omori. 2006. Acquisition of learning processing in a navigation task using a functional parts combination model. *Systems and Computers in Japan* **37**:4, 64-76. [Crossref]

196. Mai Bando, Hiroaki Nakanishi. 2006. A Stable Approach for Modular Learning and its Application to Autonomous Aero-Robot. *Journal of Robotics and Mechatronics* **18**:1, 44-50. [Crossref]

197. Mehdi Khamassi, Louis-Emmanuel Martinet, Agnès Guillot. Combining Self-organizing Maps with Mixtures of Experts: Application to an Actor-Critic Model of Reinforcement Learning in the Basal Ganglia 394-405. [Crossref]

198. Shuhei Nishida, Kazuo Ishii, Tetsuo Furukawa. An Online Adaptation Control System Using mnSOM 935-942. [Crossref]

199. Wai-Tat Fu, John R. Anderson. 2006. From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General* **135**:2, 184-206. [Crossref]

200. J. Takeuchi, O. Shouno, H. Tsujino. Connectionist Reinforcement Learning with Cursory Intrinsic Motivations and Linear Dependencies to Multiple Representations 54-61. [Crossref]

201. Stefan Schaal, Nicolas Schweighofer. 2005. Computational motor control in humans and robots. *Current Opinion in Neurobiology* **15**:6, 675-682. [Crossref]

202. Rainer W. Paine, Jun Tani. 2005. How Hierarchical Control Self-organizes in Artificial Adaptive Systems. *Adaptive Behavior* **13**:3, 211-225. [Crossref]

203. Michael E. Hasselmo. 2005. A Model of Prefrontal Cortical Mechanisms for Goal-directed Behavior. *Journal of Cognitive Neuroscience* **17**:7, 1115-1129. [Abstract] [PDF] [PDF Plus]

204. Nicole Malfait, Paul L. Gribble, David J. Ostry. 2005. Generalization of Motor Learning Based on Multiple Field Exposures and Local Adaptation. *Journal of Neurophysiology* **93**:6, 3327-3338. [Crossref]
205. Mehdi Khamassi, Loïc Lachèze, Benoît Girard, Alain Berthoz, Agnès Guillot. 2005. Actor–Critic Models of Reinforcement Learning in the Basal Ganglia: From Natural to Artificial Rats. *Adaptive Behavior* **13**:2, 131-148. [Crossref]
206. J. Dowling, E. Curran, R. Cunningham, V. Cahill. 2005. Using Feedback in Collaborative Reinforcement Learning to Adaptively Optimize MANET Routing. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **35**:3, 360-372. [Crossref]
207. Jean-Arcady Meyer, Agnès Guillot, Benoît Girard, Mehdi Khamassi, Patrick Pirim, Alain Berthoz. 2005. The Psikharpax project: towards building an artificial rat. *Robotics and Autonomous Systems* **50**:4, 211-223. [Crossref]
208. S.L. Waslander, G.M. Hoffmann, Jung Soon Jang, C.J. Tomlin. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning 3712-3717. [Crossref]
209. Masafumi Okada, Daisuke Nakamura, Yoshihiko Nakamura. 2005. Self-organizing Symbol Acquisition and Motion Generation based on Dynamics-based Information Processing System. *Transactions of the Japanese Society for Artificial Intelligence* **20**, 177-187. [Crossref]
210. A.A. Safavi, A. Khayatian, A. Aminzadeh, Y.M. Talukder, M.H. Shaeed, H.J.C. Huijberts. 2005. A STABLE REAL-TIME OPTIMAL MULTIPLE-MODEL BASED CONTROL OF A NONLINEAR PROCESS. *IFAC Proceedings Volumes* **38**:1, 889-894. [Crossref]
211. Jun Tani, Masato Ito, Yuuya Sugita. 2004. Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. *Neural Networks* **17**:8-9, 1273-1289. [Crossref]
212. Rainer W. Paine, Jun Tani. 2004. Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural Networks* **17**:8-9, 1291-1309. [Crossref]
213. Nicolas Schweighofer, Kenji Doya, Shinya Kuroda. 2004. Cerebellar aminergic neuromodulation: towards a functional understanding. *Brain Research Reviews* **44**:2-3, 103-116. [Crossref]
214. Masahiko Haruno, Daniel M. Wolpert, Mitsuo Kawato. 2003. Hierarchical MOSAIC for movement generation. *International Congress Series* **1250**, 575-590. [Crossref]
215. Kazuyuki Samejima, Kenji Doya, Mitsuo Kawato. 2003. Inter-module credit assignment in modular reinforcement learning. *Neural Networks* **16**:7, 985-994. [Crossref]
216. Yasuhiro Wada, Yasuhiro Kawabata, Shinya Kotosaka, Kenji Yamamoto, Shigeru Kitazawa, Mitsuo Kawato. 2003. Acquisition and contextual switching of multiple internal models for different viscous force fields. *Neuroscience Research* **46**:3, 319-331. [Crossref]
217. Daniel M. Wolpert, Kenji Doya, Mitsuo Kawato. 2003. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* **358**:1431, 593-602. [Crossref]
218. Nedialko I. Krouchev, John F. Kalaska. 2003. Context-Dependent Anticipation of Different Task Dynamics: Rapid Recall of Appropriate Motor Skills Using Visual Cues. *Journal of Neurophysiology* **89**:2, 1165-1175. [Crossref]
219. Yu Hiei, Takeshi Mori, Shin Ishii. Self-organized Reinforcement Learning Based on Policy Gradient in Nonstationary Environments 367-376. [Crossref]
220. Johane Takeuchi, Osamu Shouno, Hiroshi Tsujino. Modular Neural Networks for Model-Free Behavioral Learning 730-739. [Crossref]
221. T. Taniguchi, T. Sawaragi. Adaptive Organization of Generalized Behavioral Concepts for Autonomous Robots: Schema-Based Modular Reinforcement Learning 601-606. [Crossref]
222. H. Fujita, Shin Ishii. Model-based reinforcement learning for a multi-player card game with partial observability 467-470. [Crossref]
223. T. Sawada, T. Takagi, Y. Hoshino, M. Fujita. Learning behavior selection through interaction based on emotionally grounded symbol concept 450-469. [Crossref]