

Selecting Optimal Behaviors Based on Contexts

Kenji Doya^{*123}, Norikazu Sugimoto¹²³, Daniel Wolpert⁴, and Mitsuo Kawato¹³

¹ATR Human Information Science Laboratories

²CREST, Japan Science and Technology Corporation

³Nara Institute of Science and Technology

⁴Sobell Dept. of Motor Neuroscience, University College London

*2-2-2 Hikaridai, Seika, Soraku, Kyoto 619-0288, Japan

Phone: +81-774-95-1251; Fax: +81-774-95-1259

E-mail: doya@atr.co.jp

1. Introduction

In recent years the motor system has been implicated in many traditionally non-motor domains. One important idea is that the motor system is used during the perception of the action of others. For example, studies of speech perception and production have suggested a critical involvement of the articulatory control systems in perception of speech¹. The proposal is that other's actions are decoded by activating ones own action system at a sub-threshold level and that there appear to be special neural mechanism for decoding such information. Recent findings of 'mirror neurons,' which fire both during execution and observation of the same goal-directed movements, have started a debate about how neural networks responsible for motor control of one's own body can also be used to facilitate interpretation of another's movements². Human neuroimaging and magnetic stimulation studies have also shown that the areas associated with action are also active during imitation and observation³⁻⁶. Moreover, pre-motor systems are activated when subjects view manipulable tools or even action verbs^{7,8}. Such studies have brought the motor system to the forefront in the investigation of action interpretation. Evidence suggests that the mechanisms underlying motor imitation may be a critical step toward more sophisticated ways of communication, leading to the use of language^{9,10}. Here we propose a concrete computational framework that clarifies how motor control modules, combined with sensory prediction modules, can be used for learning and control of movements, as well as for imitation, cooperation, and communication¹¹.

The effortless ease with which we move masks the true complexity of the control processes involved. This is clearly evident when we try to build machines to perform human control tasks. While computers can now beat grandmasters at chess, no computer can yet control a robot to manipulate a chess piece with the dexterity of a six-year-old child. For example, during even a simple reaching movement our brain has to deal with nonlinear and nonstationarity properties of the muscles, multi-link dynamics of the arm, and variety of objects to be manipulated. Furthermore, the reach can have multiple goals, depending on

where and what we are reaching for. In order to deal with such complexity of the body dynamics and the variety of the movement goals, it has been proposed that the CNS uses a modular organization in the motor control system¹².

We proposed "modular selection and identification for control (MOSAIC)" architecture^{13,14} as a computational framework for modular learning and control. Its key idea is to use the relative goodness of predictions by multiple "forward models", are used as the "responsibility signal" for selecting multiple controllers. The effectiveness of MOSAIC architecture has been tested both in supervised learning¹⁵ and reinforcement learning¹⁶ paradigms. In this study, we present "combinatorial model-based reinforcement learning (CMRL)" architecture that efficiently combines multiple forward models, reward models, and controllers. We further consider how such architecture can be used for understanding of another's movement and more in general for communication and cooperative behaviors.

2. MOSAIC Architecture for Control

We now formulate our problem in the framework of reinforcement learning (RL)¹⁷. The agent (brain of a human, CPU of a robot, etc.) monitors the state x of the environment (including the body and external objects) and send an action (motor command) u to the environment. The state of the environment changes with a certain dynamics

$$\dot{x}=f(x,u).$$

The achievement of the goal, or any cost incurred, is signaled to the agent in the form of scalar reward

$$r=R(x,u).$$

The task for the agent is to find the policy, i.e., state-dependent action selection probability

$$P(u|x),$$

that maximizes the reward acquired in a long run. Learning is based on the prediction of future reward in the form of "value function"

$$V(x) = E[\sum_{t=0}^{\infty} \exp[-t/\gamma] r(t) dt | x(0)=x],$$

where γ is the discount factor of future reward prediction. The essential signal for learning the value function $V(x)$ as well as the policy $P(u|x)$ is the "temporal difference (TD) error"

$$\delta(t) = r(t) + \dot{V}(t) - 1/\gamma V(t),$$

which represents the difference between the predicted reward and the actual outcome¹⁷.

What makes reinforcement learning difficult in realistic environments is that the effective dynamics of the environment changes due to the nonlinearity and nonstationarity. Furthermore, reward can also change in time, depending on the need of the agent. Thus, the environmental dynamics is in general given by

$$\dot{x} = f(x, u, C),$$

where C is the context the agent is in, and the reward is given by,

$$r = R(x, u, T),$$

where T represents a particular target of action.

2.1 Combinatorial Model-based Reinforcement Learning (CMRL)

Figure 1 illustrates a version of MOSAIC architecture, “combinatorial model-based reinforcement learning (CMRL),” that can deal with multiplicity of dynamic context and goals. A defining feature of the architecture is that it utilizes arrays of forward models $f_i(x, u)$ that predicts environmental dynamics, reward models $r_j(x, u)$ that predicts the reward signal, and reinforcement learning controllers c_k that implement policies $P_k(u|x)$ based on the value functions $V_k(x)$ ¹⁶. Multiple contexts and tasks are recognized in the form of “responsibility signal” α_i^f and α_j^r , based on the accuracy of predictions of the forward and reward models. Among the array of controllers, the one best fits the current context and task is selected on the basis of the TD error δ_{ijk} which is a measure of compatibility of the forward model f_i , the reward model r_j , and the controller c_k .

Here we denote the predictions of the forward models as x_i and define the responsibility signal for the forward models as the posterior probability under Gaussian error assumption

$$\alpha_i^f = P(i|x) \quad P(x|i) = \exp(-||x-x_i||^2/\sigma_f^2).$$

The responsibility signal for the reward models are defined similarly from the differences between the predicted rewards r_j and the actual reward r

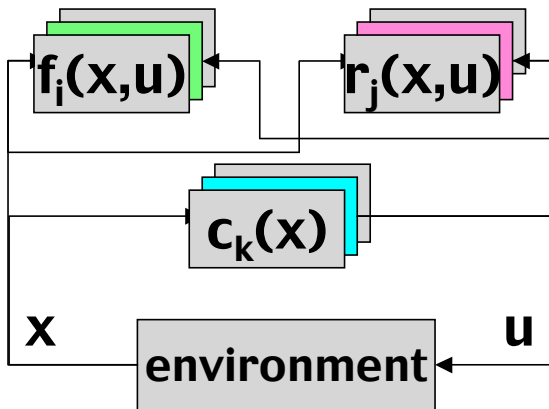


Figure 1: Combinatorial model-based reinforcement learning (CMRL) architecture.

$$\alpha_j^r = P(j|r) \quad P(r|j) = \exp(-|r-r_j|^2/\sigma_r^2).$$

The consistency between the forward model f_i , reward model r_j , and the controller c_k at state $x(t)$ can be measured by the TD error

$$\delta_{ijk}(t) = r_j(t) + \partial V_k / \partial x x_i(t) - 1 / \partial V_k(x(t)).$$

Thus the responsibility signal for the controllers are given by

$$\alpha_k^c = P(k|x, x, r) \quad \alpha_{ij}^c = \alpha_i^f \alpha_j^r \exp(-|\delta_{ijk}(t)|^2 / \sigma_c^2).$$

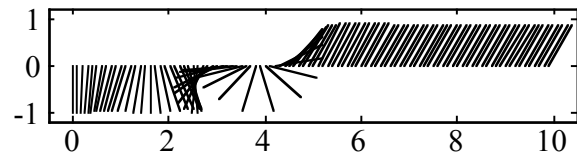
It is possible to avoid computing α_{ijk} for all possible triplets of forward models, reward models, and controllers. A simple approximation is to use the weighted prediction

$$\dot{x}^*(t) = \alpha_i \alpha_i^f(t) x_i(t)$$

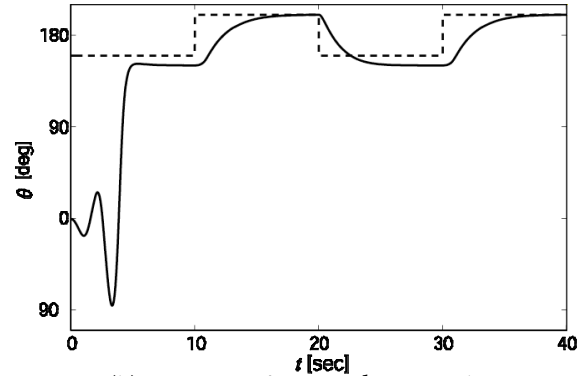
and

$$r^*(t) = \alpha_{0j} \alpha_j^r(t) r_j(t)$$

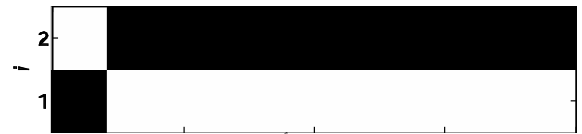
for computing the TD error for the controller c_k



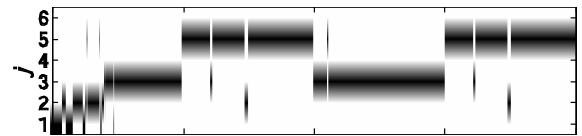
(a) swing-up behavior



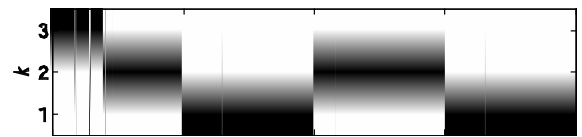
(b) trajectory for switching goals



(c) responsibility α_i^f for forward models



(d) responsibility α_j^r for reward models



(e) responsibility α_k^c for controllers

Figure 2: Swing-up control of a pendulum with switching reward functions.

$$\dot{V}_k(t) = r^*(t) + \partial V_k / \partial x \dot{x}^*(t) - 1 / \partial V_k(x(t))$$
 and then use it for computation of responsibility signal

$$\dot{V}_k^c = \exp(-|\dot{V}_k(t)|^2 / \dot{V}_k^c{}^2).$$

2.2 Simulation: Swing-up Control of a Pendulum

We tested the control performance of the CMRL architecture in a task of swinging up a weakly-powered pendulum^{16,17}. The state is $x=(\theta, \dot{\theta})$ and the action was $u=T/T_{limit}$ where θ is the angle of the pendulum and T is the torque. The maximal torque T_{limit} is small so that it is necessary to make preparatory swings to swing up the pendulum. The reward function is defined as

$$R(x,u) = \cos(\theta - \theta^*) - 0.1u^2.$$

The peak of the reward function θ^* was 20 degrees either to the left or right of the inverted position.

The CMRL model consisted of two linear forward models, six quadratic reward models, and three RL controllers. The forward models and reward models were trained on-line, while the controllers were updated off-line after each trial that lasted for 5 seconds.

Figure 2(a) shows an example of swing-up pattern after 500 learning trials. In this trial, the reward functions are switched every five seconds, as shown by the dashed line in Figure 2(b). The solid line in Figure 2(b) shows the trajectory of swing-up and subsequent swinging movement due to the changes in the reward function. Two forward models predicted the pendulum dynamics of the bottom (f_1) and top (f_2) halves of the state space, and they were appropriately selected according to the change of the pendulum state (Figures 2c).

Among the six reward models, two (r_1 and r_2) covered the bottom half of the state space and two others (r_3 and r_5) covered the top half, each for different setting of the reward function. As shown in Figure 2(d), the responsibility signal for the reward models changed according to the changes in the reward, which in turn resulted in the selection of appropriate controllers (Figure 2e).

3. MOSAIC for Communication

In the above setup, different reward functions were given externally by the environment. When we perform complex behaviors, however, we subjectively set certain sub-goals in order to achieve the final goal, which is given as the reward from the environment. In order to allow selection of sub-goals by the agent, we introduce a hierarchical organization to the MOSAIC architecture.

3.1 Hierarchical MOSAIC

A key principle in hierarchical MOSAIC is that the situation that is well predicted by a single forward model f_i is considered as a single context C , which is regarded as an abstract state of the upper level dynamics. The upper-level state can be either a discrete index i , or a continuous vector \dot{V}_i^c representing the probability over the indices. The action for the upper level is the target T , in the form of the index j of the reward model, or a prior for the responsibility signal \dot{V}_j^c for reward model selection. The upper-level policy $P(T|C)$ specifies which target T is to be selected under given context C .

This architecture is not only useful for complex motor control tasks, but also for interpretation of

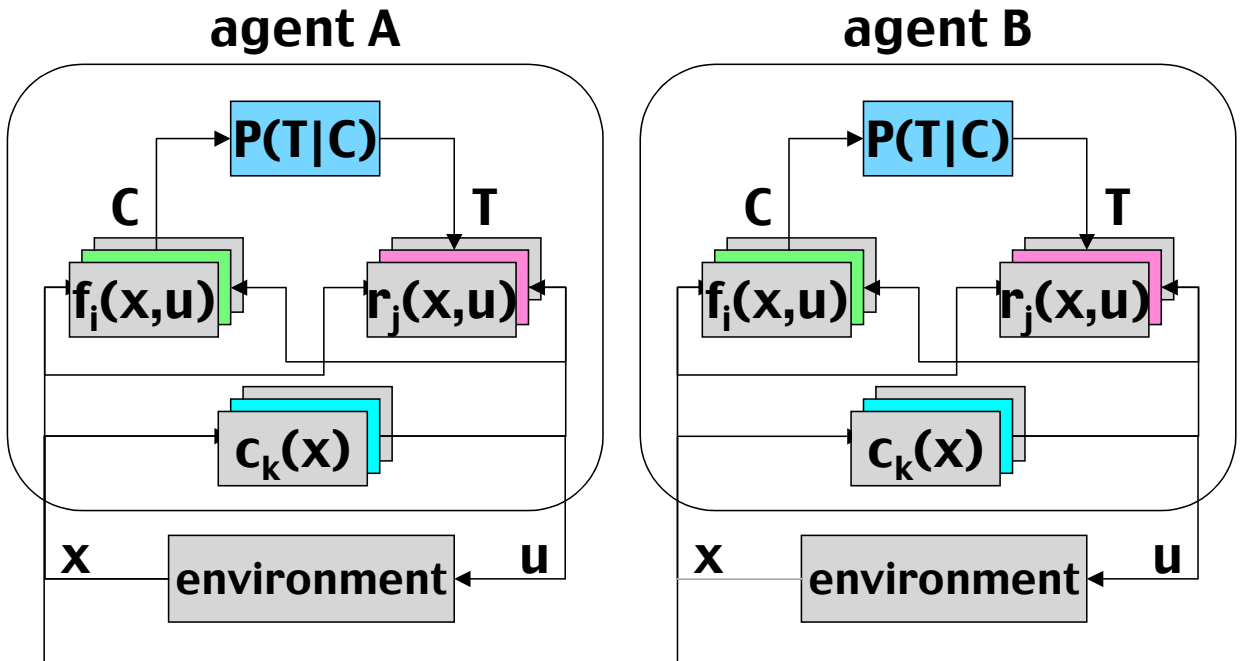


Figure 3: Hierarchical MOSAIC architecture for communication.

another agent's actions for the sake of imitation, cooperation, and communication.

1. **Control:** The upper level learns policy $P(T|C)$ to maximize the reward from the environment. In the lower level, while the reward model responsibility \square_j^r is specified by the upper level, the forward model responsibility \square_i^f is given by the fit with the current environmental dynamics. The controller that is most consistent with the forward and reward models is selected using the TD error \square_{ijk} .
2. **Imitation:** Now agent B tries to imitate the behavior of agent A by observing its state trajectory $x^A(t)$. Under the assumption that two agents A and B share roughly the same set of internal models, agent B can predict agent A's action $u_k^A(t) = c_k(x^A(t))$, and resulting state change $x_{ik}^A(t) = f_i(x^A(t), u_k^A(t))$. This can then be used to estimate which controller-forward model pair best fits the observed trajectory, and also which reward model is likely to be used with the criterion of minimal TD error \square_{ijk} . Thus the agent B can estimate context C^A and task goal T^A of agent A and can learn to mimic the upper level policy $P(T^A|C^A)$ of agent A.
3. **Cooperation:** In order for agent B to cooperate with agent A, its action should be based not only on the state of the environment x , but also on the internal state of agent A, including its recognition of the context C^A , and its current target T^A . The same mechanism as in imitation can be used to estimate C^A and T^A of agent A. The task for agent B in cooperation, however, is not simply to mimic the same policy $P(T^A|C^A)$ as agent A, but to select its own target T to complement agent A's behavior. Thus agent B should learn a policy $P(T|C, C^A, T^A)$ in the extended upper-level state. The same mechanism is required for agent A for reciprocal cooperation.
4. **Communication:** In general communication, two agents may not have access to the same environment and they can have different goals. The task for agent A in communication is to let agent B know the context C^A he is in and the target T^A he has in mind through observation of its motor outcome x^A , and thereby affect the target T^B selected by agent B. The appropriate policy for agent A depends on the set of internal models agent B has, and whether agent B is cooperative or adversary to agent A.

An important point in the use of hierarchical MOSAIC is that, from observation of physical state x , upper-level context C and target T are extracted. This allows, for example, imitation of movements despite differences in the physical parameters of each agent.

3.2 Imitation of Swinging Pendulum

In order to test the above concepts, here we take an example of imitation of swinging pendulum. Agent A is the demonstrator and agent B is the imitator, but the maximal output torque

T_{limit} of agent B is smaller than that of agent A. Thus agent B cannot exactly replicate agent A's movement.

In Figure 4(a), the dashed line shows the trajectory x^A demonstrated by agent A. It swings left and right with 60 degrees amplitude around the inverted position.

From this trajectory, agent B estimates the action $u_k^A = c_k(x^A)$ for the three controllers and then predict the state change $x_{ik}^A = f_i(x^A, u_k^A)$ with the two forward models. The consistency of these predictions to the observed trajectory x^A is given by

$$\square_{ik}^A P(x|i,k) = \exp(-||x^A - x_{ik}^A||^2 / \square_i^2).$$

The responsibility signals for the forward models and controllers are then given by

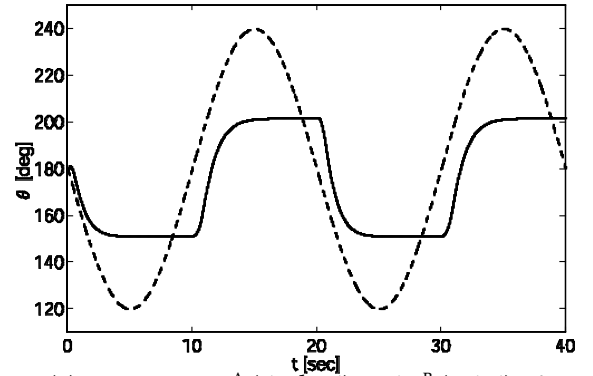
$$\begin{aligned} \square_i^A &= \square_k \square_{ik}^A \\ \square_k^A &= \square_i \square_{ik}^A \end{aligned}$$

Figure 2(b) shows the time course of estimated controller responsibility signal \square_k^A which successfully estimates two controllers with equilibriums at left and right. The solid line in Figure 2(a) shows the resulting trajectory x_B of agent B. Note that the amplitude is smaller than the demonstrated trajectory x_A . This is because agent B imitated the selection of control modules, which reflect the physical properties of agent B, instead of just replicating the trajectory of agent A.

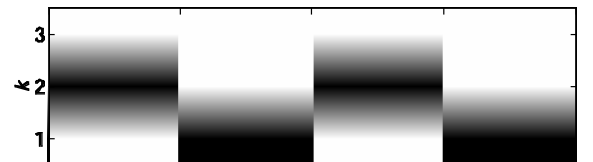
When agent B used an ordinary tracking controller with the demonstrated trajectory x^A as the target, the pendulum fell off because of the limit of its output torque.

In order to estimate the responsibility \square_j^r for the reward models, we compute the TD error

$$\square_{jk}^r(t) = r_j(t) + \partial V_k / \partial x x^A(t) - 1 / \square V_k(x^A(t)),$$



(a) trajectories x^A (dashed) and x^B (solid) of agents A and B, respectively.



(b) responsibility \square_k^A estimated from trajectory x^A of agent A and internal models of agent B.

Figure 4: Imitation of movement under different physical constraint.

which represents the consistency of the reward model r_j and the controller c_k with the observed trajectory x^A . Thus the responsibility signal for the reward models is given by

$$\square_j^{Ar} \square_k \square_k^c \exp(-|\square_{jk}^A(t)|^2 / \square_c^2).$$

In the current example, we simply used the time course of \square_k^{Ac} as the responsibility signal for controller selection by agent B. It is possible, however, to learn the upper level policy $P(T|C)$ based on the estimates of the responsibilities \square_i^{Af} for the forward models and \square_j^{Ar} for the reward models.

4. Discussion

We demonstrated in this study that the MOSAIC architecture, which was originally developed for motor control, can be helpful also in estimating the intention of another agent in the form of responsibility signals for the controllers and the reward models. Although the idea that the motor system is helpful for perception and communication has been around for many years, this is to our knowledge the first demonstration of how that could be realized in realistic motor control and communication tasks.

A critical question now is whether such a computational mechanism can be implemented in our brain. Previous theoretical models suggest that, while cerebellum is specialized in supervised learning to acquire internal models of body and external environment, the cerebral cortex is specialized in unsupervised learning, including modular decomposition of sensory-motor information^{18,19}. Thus one possible implementation of MOSAIC architecture is that while multiple forward models and controllers are located in the cerebellum, the cortical areas that receive cerebellar outputs are the locus of adaptive decomposition and selection of modules. If that is the case, the cortical neurons representing the responsibility signals are activated during both execution and observation of goal-directed movements, as do the mirror neurons. Whether the proposed computational scheme indeed is the one that is used in our brain remains to be tested by neural recording and brain imaging studies.

References

1. Liberman, A. M. & Whalen, D. H. On the relation of speech to language. *Trends Cogn Sci* **4**, 187-196 (2000).
2. Rizzolatti, G., Fogassi, L. & Gallese, V. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat Rev Neurosci* **2**, 661-70 (2001).
3. Fadiga, L., Fogassi, L., Pavesi, G. & Rizzolatti, G. Motor facilitation during action observation: a magnetic stimulation study. *J Neurophysiol* **73**, 2608-11 (1995).
4. Fadiga, L., Craighero, L., Buccino, G. & Rizzolatti, G. Speech listening specifically modulates the excitability of tongue muscles: a TMS study. *Eur J Neurosci* **15**, 399-402 (2002).

5. Grezes, J. et al. Does perception of biological motion rely on specific brain regions? *Neuroimage* **13**, 775-85 (2001).
6. Iacoboni, M. et al. Cortical mechanisms of human imitation. *Science* **286**, 2526-8 (1999).
7. Grafton, S. T., Fadiga, L., Arbib, M. A. & Rizzolatti, G. Premotor cortex activation during observation and naming of familiar tools. *Neuroimage* **6**, 231-6 (1997).
8. Martin, A., Wiggs, C. L., Ungerleider, L. G. & Haxby, J. V. Neural correlates of category-specific knowledge. *Nature* **379**, 649-52 (1996).
9. Donald, M. *Origins of the modern mind: Three stages in the evolution of culture and cognition* (Harvard University Press, Cambridge, Massachusetts, USA, 1991).
10. Rizzolatti, G. & Arbib, M. A. Language within our grasp. *Trends Neurosci* **21**, 188-94 (1998).
11. Wolpert, D. M., Doya, K. & Kawato, M. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society London, B* (2003).
12. Ghahramani, Z. & Wolpert, D. M. Modular decomposition in visuomotor learning. *Nature* **386**, 392-5 (1997).
13. Wolpert, D. M. & Kawato, M. Multiple paired forward and inverse models for motor control. *Neural Networks* **11**, 1317-1329 (1998).
14. Wolpert, D. M. & Ghahramani, Z. Computational principles of movement neuroscience. *Nat Neurosci* **3 Suppl**, 1212-7 (2000).
15. Haruno, M., Wolpert, D. M. & Kawato, M. Mosaic model for sensorimotor learning and control. *Neural Comput* **13**, 2201-20 (2001).
16. Doya, K., Samejima, K., Katagiri, K. & Kawato, M. Multiple model-based reinforcement learning. *Neural Comput* **14**, 1347-69 (2002).
17. Doya, K. Reinforcement learning in continuous time and space. *Neural Comput* **12**, 219-45 (2000).
18. Doya, K. What are the computations of the cerebellum, the basal ganglia, and the cerebral cortex. *Neural Networks* **12**, 961-974 (1999).
19. Doya, K. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Curr Opin Neurobiol* **10**, 732-9 (2000).