Hand Reaching Movement Acquired through Reinforcement Learning

Katsunari Shibata†, Masanori Sugisaka†& Koji Ito‡

†Dept. of Electrical & Electronics Engineering, Faculty of Engineering, Oita University
700 Dannoharu, Oita 870-1192, JAPAN
‡Dept. of Computational Intelligence & Systems Science, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, JAPAN
{*shibata, msugi*}*@cc.oita-u.ac.jp, ito@dis.titech.ac.jp*

Abstract

This paper shows that a system with two-link arm can obtain hand reaching movement to a target object projected on a visual sensor by reinforcement learning using a layered neural network. The reinforcement signal, which is an only signal from the environment, is given to the system only when the hand reaches the target object. The neural network computes two joint torques from visual sensory signals, joint angles, and joint angular velocities considering the arm dynamics.

It is known that the trajectory of the voluntary movement of human hand reaching is almost straight, and the hand velocity changes like bell-shape. Although there are some exceptions, the properties of the trajectories obtained by the reinforcement learning are somewhat similar to the experimental result of the human hand reaching movement.

Key Words: Direct-Vision-Based Reinforcement Learning, Neural Network, Hand-Eye Coordination, Reaching Task, Trajectory Planning

1 Introduction

It has been proposed that the visual sensory signals are put into a neural network directly and the network is trained to output appropriate motion signals by reinforcement learning[1]. The continuous state space can be formed adaptively and purposively through the learning. It shows the possibility that the reinforcement learning is useful not only for the motion planning, but for the total functions from sensors to motors, including recognition, attention, and so on. This is called Direct-Vision-Based Reinforcement Learning.

It has been shown that hand-eye coordination can be obtained by the combination of reinforcement learning and neural network in a robot arm reaching task[2]. It has been realized only by adding the joint angles as input signals in Direct-Vision-Based Reinforcement Learning. However, the dynamics of the arm was not introduced, but the joint angular velocities were the output of the neural network.

In this paper, the dynamics of the arm is introduced. It means that the joint angular velocities are also the input signals to the neural network, and the output signals are the joint torques. Finally, it is shown that the trajectories and tangential velocity curve of the hand after learning have somewhat similar properties to the experimental result of the human hand reaching movement.

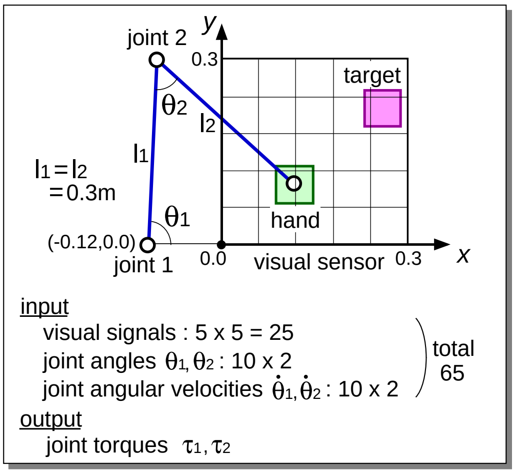

Figure 1: The robot hand-reaching task.

2 Task Setting

2.1 General Setting

Here the setting of the task as shown in Fig. 1 is described. The visual sensor has $5 \times 5 = 25$ cells and the output of each cell is the area ratio occupied by the target or the robot's hand against its receptive field. Below here, the left-bottom corner of the visual sensor is supposed to be the origin. The size of each visual cell is 0.06×0.06, and the size of the target and hand is also the same. The target and the hand cannot be distinguished with each other on this visual sensor.

The target is located randomly in the range where the whole target can be caught in the visual field, i.e., $0.03 \leq x, y \leq 0.27$. The initial hand location is also chosen randomly. In the early phase of the learning, it is chosen from only around the target, and according to the progress of the learning, the range becomes wider gradually until $-0.09 \leq x, y \leq 0.39$ under the condition of $((x - 0.12)^2 + y^2) \leq (l_1 + l_2)$. So the hand sometimes cannot be caught by the visual sensor initially after some trials. The base of the arm (joint 1) is fixed at (-0.12, 0.0). The both joint angles are limited in the range of $0 \leq \theta_i \leq \pi$, and joint angular velocities are limited in the range of $-\pi \leq \dot{\theta}_i \leq \pi$. The target is fixed during one trial. But if the hand cannot reach after many time steps, the trial finishes with no reward, and in some following trials, the target is moved towards the hand gradually. The length of each link is the same as the side of the visual sensor. There is

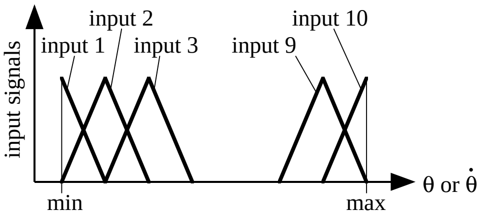

Figure 2: Localization of a continuous input signal into 10 signals.

Table 1: Parameters used in the dynamical arm model.

Parameter	link1	link2
Mi (kg)	2.0	
li (m)	0.3	
si (m)	li/2	
Ii (kg m^2)	Mi * li^2 / 3.0	
Bi (kg m^2/s)	0.4	0.2
$\tau_{max\,i}$ (N m)	8.0	4.0

no singularity and one-to-one correspondence between the joint angles and hand location exists.

The inputs of the neural network are the visual signals, the joint angles, and the joint angular velocities. Each of the joint angles and joint angular velocities is localized into 10 signals as shown in Fig. 2. The localization is useful to approximate a strong non-linear function. The total number of inputs is 65. There are two outputs which represent the joint torques, When the hand touches the target, and the hand tangential velocity is less than $0.06\sqrt{8}$, the reward 0.4 is given, When the joint angle goes over its limit, the penalty -0.4 is given. When the system obtains the reward or penalty, the trial finishes. The neural network has three layers, and the number of hidden neurons is 20. The output function of each neuron in the hidden and output layer is sigmoid function whose value range is from -0.5 to 0.5. The neural network is trained by Error Back Propagation algorithm.

2.2 Dynamical Arm Model

The differential equation that describes the arm dynamics is as follows.

$$\begin{aligned} \tau_1 &= (I_1 + I_2 + 2M_2l_1s_2cos\theta_2 + M_2(l_1)^2\ddot{\theta}_1 \\ &\quad +(I_2 + M_2l_1s_2cos\theta_2)\ddot{\theta}_2 \\ &\quad -M_2l_1s_2(2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2sin\theta_2 + B_1\dot{\theta}_1 \qquad (1) \\ \tau_2 &= (I_2 + M_2l_1s_2cos\theta_2)\ddot{\theta}_1 + I_2\ddot{\theta}_2 \\ &\quad +M_2l_1s_2(\dot{\theta}_1)^2sin\theta_2 + B_2\dot{\theta}_2 \qquad (2) \end{aligned}$$

Here M_i, l_i, s_iandI_i represent the mass, the length, the distance from the joint to the center of mass, and the rotary inertia of the link i around the joint, respectively. The parameters are shown in Table 1. The equation is numerically solved by Runge-Kutta method. The sampling time is 0.02sec in the learning period, and 0.01sec after learning.

2.3 Reinforcement Learning

The basic architecture is the Actor-Critic[3], but only one layered neural network makes both roles of Actor and Critic. The algorithm is Temporal-Smoothing (TS) based reinforcement learning. This is very similar to Temporal Difference (TD) based reinforcement learning[3]. Only the difference is that the curve of the value function along time axis becomes straight line in TS on behalf of exponential curve in TD.

Here, Adaptive Slope Method is also employed, in which the slope of the value function along time axis is changed adaptively according to the progress of the learning. The slope corresponds to the discount factor in TD-based learning. The ideal slope ΔV_{ideal} is computed as

$$\Delta V_{ideal} = (V_{max} - V_{min})/N_{max} \qquad (3)$$

where V_{max} : the upper limit of the value, here 0.4, V_{min} : the lower limit, here -0.4. For adaptability, N_{max} is computed when reaching the target as

$$N_{max}[i] = \begin{cases} N[i] & \text{if } N[i] > \beta N_{max}[i-1] \\ \beta N_{max}[i-1] & \text{otherwise} \end{cases} \qquad (4)$$

where $N[i]$: time steps to reach the target at the i-th trial, β: an attenuation factor ($0.0 < \beta < 1.0$, here 0.9996). Then by comparing the change of the actual value to this ideal one, the value at the previous time $V(t-1)$ is trained by the training signal as

$$V_s(t-1) = V(t-1) - \eta(\Delta V_{ideal} - \Delta V(t)) \qquad (5)$$

where $\Delta V(t) = V(t) - V(t-1)$, and η : a training constant. When the hand arrives at the target, or the joint angle goes over the limitation, the value is trained to be 0.4 or -0.4 respectively.

The joint torques are generated in proportion to the sum of the motion signals **m** and random numbers **rnd** as trial and error factors. The random number is uniform random number powered by 3, and the amplitude of the random number is adjusted according to the relative gain of the value function as $\Delta V/\Delta V_{ideal}$. If the gain is small, the amplitude becomes large. The motion signals **m** are trained by the training signals as

$$\mathbf{m}_s = \mathbf{m} + \zeta\mathbf{rnd}\Delta V \qquad (6)$$

where ζ : a training constant. These two learnings are processed in parallel.

3 Simulation Result

Fig. 3 shows some examples of the hand trajectory and hand tangential velocity curve after learning. Fig. 4 shows the arm configurations at the initial hand locations. From these results, it can be seen that the hand trajectory is roughly straight and the hand tangential velocity curve has one peak and is roughly bell-shaped. Some of them do not look like bell-shape exactly. The reason may be insufficient learning due to the insufficient resolution of sensory signals, insufficient hidden

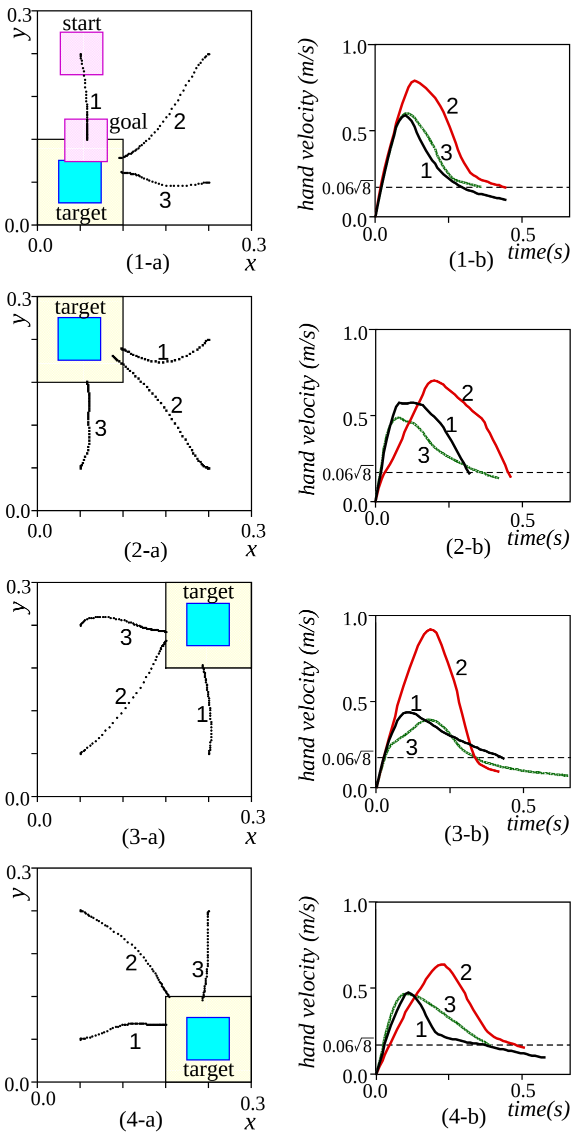

Figure 3: Examples of (a) hand trajectory and (b) hand tangential velocity curve for 12 combinations of the target and hand location after learning. The large square in (a) around the target shows the range at which the hand touches the target. The horizontal broken line in (b) shows the upper limit velocity at which the system can obtain the reward.

neurons, or so on. That is because the trajectory is far from the minimum time trajectory, even though the minimum time trajectories are expected to be obtained by reinforcement learning.

Fig. 5 shows the torques and joint angular velocities in the case of Fig. 3 (3) 2, in which the hand started from (0.06, 0.06) and the target was located at (0.24, 0.24). It can be noticed that the joint 2 torque accelerates the joint angular velocity at first and then the torque becomes negative to stop the joint. The joint 1 moves little, but the joint 1 torque changes a couple of times between positive and negative values. Fig. 6 shows the value function and hand motion vector as a function of the hand location when the target is at (0.24, 0.24) and the both joint angular velocities are 0.0. The motion vector is computed by the torque output. The peak of the value function can be seen at the target, and the hand moves towards the peak even if the other small peak exists around the origin.

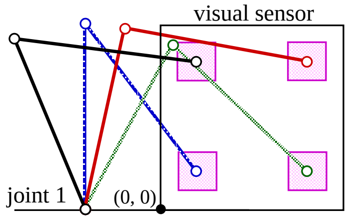

Figure 4: The relation between the hand location and the arm configuration.

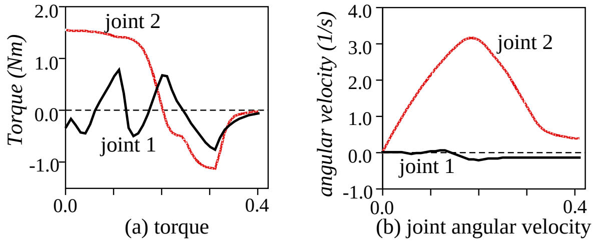

Figure 5: The joint torques and angular velocity when the trajectory is as Fig. 3(3)2.

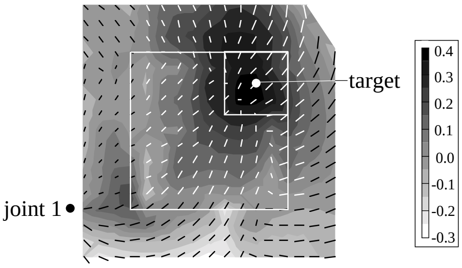

Figure 6: The value function and the hand tangential velocity vector as a function of the hand location when the target is located at the same as Fig. 3(3)2, and the initial angular velocities are both 0.0. The large square shows the visual sensor, and the small square shows the range where the hand touches the target.

Fig. 7 shows the hand trajectory when the hand is located out of the visual field initially. It also can be seen that the trajectory is smooth and close to a straight line, and the hand velocity has one peak.

As described above, the hand trajectory is almost close to the straight line, and the velocity curve has one peak in many cases. However, in some combinations of the initial hand and target location, the hand trajectory is not close to a straight line and the hand tangential velocity curve has more than one peak as shown in Fig. 8.

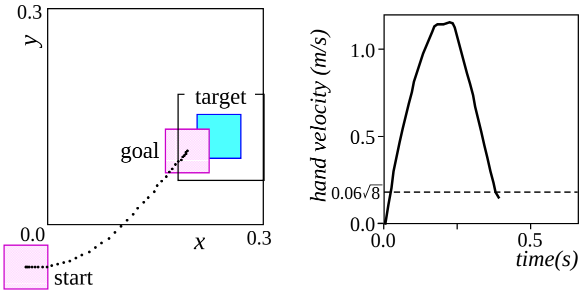

Figure 7: An example of hand movement when the initial hand location is out of the visual sensor.

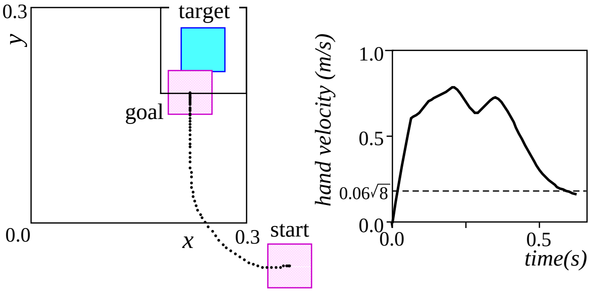

Figure 8: An example in which the hand trajectory is not similar to straight line and hand tangential velocity curve has more than one peak.

4 Comparison with Human Movement

It is well-known that in the voluntary movements of our hand to reach some target which is not located so far, the trajectory is almost straight and the velocity of the hand forms a bell shape along the time axis[4]. It is also reported that even when the subject is requested that the hand reaches the target as soon as possible, the trajectory is almost straight line and the hand velocity changes like bell-shape[5].

In general, it has been thought that the generation of human movement is composed of three processes, trajectory planning, coordination transformation from the work space to joint space, and generation of motor command considering its dynamics. Under this idea, some models of the trajectory planning have been proposed[6][4][7]. In these models the trajectory of the hand is obtained by solving the optimization problem under the cost function like minimum commanded torque change and so on. So, iterative computation is necessary every time when a new reaching is required.

When comparing the trajectory after learning in this paper with that obtained by the conventional models, it is not so closer to the human's. However, in our model, iterative computation is not necessary after the learning even though many trials (here 1,000,000 trials) for learning are necessary.

Furthermore, it is not enough to say that our model is a model of human reaching movement, due to many insufficient points, e.g., the fixed visual sensor, the uniform visual sensory cells, no impedance based on the redundant muscles, and so on. It is expected to be improved in the future.

5 CONCLUSION

The hand-reaching task was achieved by the combination of reinforcement learning and neural network. The hand trajectory and tangential velocity was roughly similar to the human's. The authors think that the possibility could be shown that reinforcement learning works in human brain to acquire the arm movement.

Acknowledgments We would like to say thank to Prof. Y. Uno, Mr. H. Kubotera and Mr. J. Izawa to give us much knowledge about the human hand reaching movement and the conventional models, and to discuss with us. A part of this research was supported by the Scientific Research Foundation of the Ministry of Education, Science, Sports and Culture of Japan (#10450165) and by "The Japan Society for the Promotion of Science" as "Biologically Inspired Adaptive Systems" (JSPS-RFTF96I00105) in "Research for the Future Program".

REFERENCES

[1] Shibata, K., Okabe, Y. & Ito, K., "Direct-Vision-Based Reinforcement Learning in "Going to an Target" Task with an Obstacle and with a Variety of Target Sizes", *Proc. of NEURAP'98*, pp. 95-102, 1998.

[2] Shibata, K., & Ito, K., "Hand-Eye Coordination in Robot Arm Reaching Task by Reinforcement Learning", *Proc. of IEEE SMC'99*, **V**, pp. V-458-463, 1999.

[3] Barto, A. G., Sutton, R. S. & Anderson C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans. of SMC*, **13**, pp. 835-846, 1983.

[4] Uno, Y., Kawato, M. & Suzuki, Y., "Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement - Minimum Torque-Change Model", *Biological Cybernetics*, **61**, pp. 89-101, 1989.

[5] Suzuki, K., & Uno, Y., "Brain Adopts the Criterion of Smoothness for Most Quick Reaching Movements", *Trans. of IEICE*, **J83-D-II**, pp.711-722, 2000. (in Japanese)

[6] Flash, T. & Hogan, N., "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model", *J. Neuroscience*, **5**, pp.1688-1703,1985.

[7] Nakano, E., et al., "Quantitative Examinations of Internal Representations for Arm Trajectory Planning, Minimum Commanded Torque Change Model", *J. Neurophysiology*, **81**, No. 5, pp. 2140-2155, 1999.