Research Article

Decision Making in Reinforcement Learning Using a Modified Learning Space Based on the Importance of Sensors

Yasutaka Kishima, Kentarou Kurashige, and Toshihisa Kimura

Muroran Institute of Technology, 27-1 Mizumoto, Hokkaido, Muroran 0508585, Japan

Correspondence should be addressed to Kentarou Kurashige; kentarou@csse.muroran-it.ac.jp

Received 15 March 2013; Accepted 21 May 2013

Academic Editor: Guangming Song

Copyright © 2013 Yasutaka Kishima et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many studies have been conducted on the application of reinforcement learning (RL) to robots. A robot which is made for general purpose has redundant sensors or actuators because it is difficult to assume an environment that the robot will face and a task that the robot must execute. In this case, Q-space on RL contains redundancy so that the robot must take much time to learn a given task. In this study, we focus on the importance of sensors with regard to a robot's performance of a particular task. The sensors that are applicable to a task differ according to the task. By using the importance of the sensors, we try to adjust the state number of the sensors and to reduce the size of Q-space. In this paper, we define the measure of importance of a sensor for a task with the correlation between the value of each sensor and reward. A robot calculates the importance of the sensors and makes the size of Q-space smaller. We propose the method which reduces learning space and construct the learning system by putting it in RL. In this paper, we confirm the effectiveness of our proposed system with an experimental robot.

1. Introduction

In recent years, reinforcement learning (RL) [1] has been actively studied, and many studies on its application to robots have been conducted [2–4]. A matter of concern in RL is the learning time. In RL, information from sensors is projected onto a state space. A robot learns the correspondence between each state action in the state space and determines the best correspondence. When the state space expands according to the number of sensors, the number of correspondences learned by the robot is also increased. In addition, the robot needs considerable much experience in each state to perform a task. Therefore, learning the best correspondence becomes time-consuming.

To overcome this problem, many studies have investigated accelerated RL [5–15] for which there are two approaches: a multirobot system and autonomous construction of the state space. In the former approach, multiple robots exchange experience information [5–9], so that each robot augments its own knowledge. Therefore, in this system robots can find the best correspondence between each state

and action faster than an individual robot in a single-robot system. In addition Nishi et al. [10] proposed a learning method in which a robot learns behavior through observations of the behavior of other robots, constructing its own relationships between state and behavior. However, in this approach, a robot needs other robots with whom to exchange experience information, and hence, if there are no additional robots in the system, this approach becomes irrelevant. We focus on the state construction of a single robot.

In contrast to the above approaches, in the approach that applies autonomous state space construction [11–16], a single robot is sufficient. The robot constructs a suitable state space based on its experience. Moreover, it can reduce the state space and learn correspondences faster. However, in the studies on this approach, all the sensors installed in the robot were considered to be equally important, and their number of states was the same. The installed sensors, which influence how well a robot executes a task, can be divided into important and unimportant sensors according to the task to be performed. However, in this approach, the robot has

to learn using unnecessary inputs because all the sensors are considered equally important. For example, Takahashi et al. [16] proposed a state that is constructed autonomously from state space by incremental state segmentation. In this method, the division rule of state is applied to the consecutiveness of the sensor data. Ishiguro et al. [12] proposed a state construction method using empirically obtained perceivers (EOPs). These methods are not focused on the importance of each sensor for performing a task. In fact, although the sensors installed on a robot have varying levels of importance in terms of performing a task, few studies focused on this aspect. We focus on the importance of a sensor for a particular task and propose a novel efficient learning method.

In this paper, we propose a system in which the robot constructs a temporary Q-space for decision making based on which sensors are considered important for the execution of a particular task, which facilitates high-speed learning. Since very important sensors affect the performance of a task significantly, they should sense the environment circumstantially. Thus, the number of their states is increased. On the other hand, since less important sensors do not affect the performance of a task, they may sense their environment less exactly. Thus, the number of their states, which is determined based on the importance of the sensors, is decreased. In this study, the importance of the sensors is defined as the correlation between the sensor value and the reward.

A temporary Q-space is constructed from the Q-space of the robot based on the importance of the sensors. The number of states of the Q-space is the maximum number that the sensors can describe. A Q-space is reduced by merging Q-values according to the number of the states of the sensors. Using the reduced Q-space, the robot can efficiently select an action, which is based on more information because the Q-values of low importance are merged. Therefore, the robot can learn correspondences using fewer experiences.

This system is effective for a variety of tasks. When this method is implemented, the amount of information that the robot requires in order to learn correspondences is reduced. As a result, when our proposed system is applied, a robot can learn correspondences faster than when an ordinary RL is applied.

2. Concept of Importance of Sensors

To select the sensors that are important to a certain task, a robot needs to measure the importance of each sensor. We focus on the correlation of the sensor value and the reward, which is specific to each task, as the measure of the importance of each sensor for a task. For example, in a garbage collection task, a robot is expected to approach a garbage heap and lift it. For this task, the reward is expressed by the distance between the robot and garbage heap and increases as the robot moves closer to the garbage heap. This implies that there is a correlation between the reward and the distance between the robot and garbage heap.

We show in Figure 1 an outline of the determination of the importance of the sensors where two types of sensor are installed on the robot. Via its sensors, the robot recognizes its

environment, which is expressed as a group of all the sensor values. The robot collects the sensor value of each sensor and the reward of the task to be performed and then determines the correlation between them. In Figure 1, the robot conducts this determination for sensors 1 and 2. The robot estimates the importance of the sensors according to the two types of correlation between the sensor value and reward: negative and positive.

Very important sensors affect the performance of a task. Therefore they should sense their environment circumstantially, and thus the number of their states is increased. On the other hand, less important sensors do not affect the performance of a task, and therefore they may sense the environment less exactly, and thus, the number of their states is decreased. The number of states is therefore determined based on the importance of the sensors.

3. Decision Making in Reinforcement Learning Using a Q-Space Based on the Importance of Sensors

3.1. Outline of the Proposed System. We show our proposed system in Figure 2. In the figure, the proposed system is divided into two stages. The first stage constitutes the proposed method whereby the robot determines the importance of its sensors. The next stage is RL.

In the first stage, the robot calculates the importance of the sensors for a task based on the correlations between each sensor and reward. The robot first estimates each sensor value and the reward and then calculates the coefficient of the correlation between them. Finally, it determines the important sensors based on this coefficient of correlation.

In RL, a robot learns the actions that are suitable for each state. This stage consists of an action evaluation element, wherein a temporary Q-space is constructed based on the determination of important sensors, and an action selection element. In the action evaluation element, each pair of state and action is evaluated and updated. A state contains the value of all the sensors. The robot constructs a temporary Q-space, by adding to it only those sensors that have been determined to be important. In the action selection element, the robot selects the action for a state recognized by the sensor based on the temporary Q-space.

We show the workflow of our proposed system in Figure 3. This workflow is executed by the robot for each action. We define this flow as one trial.

3.2. Determination of the Importance of Sensors. In the proposed method, the robot determines the importance of its sensors based on the correlation between each sensor value and reward, which it experiences. The robot stores each sensor value and the averaged reward as a list called a knowledge list, an example of which is shown in Figure 4. When a robot identifies a state that is not in the list, it adds it to the list. Then, the robot calculates the averaged reward and adds it to the list. On the other hand, when a recognized state is already in the list, it calculates the averaged reward and updates the list.

FIGURE 1: Outline of determination of importance.

FIGURE 2: Outline of the proposed system.

A state s_i is defined as (1), where i is the state ID in Figure 4, $e_{i,j}$ is the value of sensor j , and E_j is a group of values that describe a sensor:

$$s_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,j}, e_{i,n} \mid e_{i,1} \in E_1, e_{i,2} \in E_2, \dots, e_{i,j} \in E_j, e_{i,n} \in E_n\}. \quad (1)$$

In this study, the rewards for the state experienced by the robot are weight averaged. Weighted averaging gives a greater weight to more recently obtained rewards. The averaged reward in state s_i is denoted by $r'_t(s_i)$ and it is updated as (2), where $r_t(s_i)$ is the reward obtained by the robot at time t :

$$r'_t(s_i) \leftarrow r_t(s_i) + \alpha_{\text{ave}}(r_t(s_i) - r'_{t-1}(s_i)). \quad (2)$$

The robot calculates the importance of each sensor based on the knowledge list. We use an equation for multiple coefficient correlation to calculate the sensor's level of importance.

In multiple coefficients, each partial regression coefficient represents the importance of each sensor. A sensor with a higher regression coefficient has a higher importance level.

The multiple regression equation is defined by (3), where $a_{t,1}, a_{t,2}, \dots, a_{t,j}$ are the regression coefficients for each sensor, $r'_t(t, s)$ is the averaged reward in state s , and a_0 is the constant term:

$$r'_t(s_i) = a_1 e_{i,1} + a_2 e_{i,2} + \dots + a_i e_{i,n} + a_0. \quad (3)$$

A robot needs to calculate the regression coefficients of each sensor to calculate its importance. Each regression coefficient $a_1, a_2, \dots, a_n$ is the solution resulting from the multiple simultaneous equation (4). Here, b_{ij} , calculated by (5), is the covariance of sensor i and sensor j , and b_i^2 , calculated by (6), is the variance of sensor i . The average of sensor i values is calculated by (7), and the $\bar{e}_i$ and $\bar{e}_j$ in (5) are

FIGURE 3: Flow of the proposed system.

State ID	Sensor value of sensor 1	Sensor value of sensor 2	...	Sensor value of sensor n	Averaged reward $r'(s_i)$
1	10	56	...	111	45
2	23	45	...	123	23
⋮					
m	90	10	...	15	132

FIGURE 4: List for calculating correlation.

calculated by (7). b_{ir} , calculated by (8), is the covariance of sensor i and reward r , and r' , calculated by (9), is the average value of the averaged reward:

$$\begin{aligned}
 a_1 b_1^2 + a_2 b_{12} + \dots + a_n b_{1n} &= b_{1r}, \\
 a_1 b_{12} + a_2 b_2^2 + \dots + a_n b_{2n} &= b_{2r}, \\
 &\vdots
 \end{aligned} \tag{4}$$

$$a_1 b_{1n} + a_2 b_{2n} + \dots + a_n b_n^2 = b_{nr},$$

$$b_{ij} = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)(e_{j,k} - \bar{e}_j), \tag{5}$$

$$b_i^2 = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)^2, \tag{6}$$

FIGURE 5: Variation property.

$$\bar{e}_j = \frac{1}{m} \sum_{k=1}^m e_{j,k}, \tag{7}$$

$$b_{ir} = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)(r_k - \bar{r}), \tag{8}$$

$$\bar{r}' = \frac{1}{m} \sum_{k=1}^m r'(s_k). \tag{9}$$

3.3. Determination of the Number of States of Each Sensor. The number of states is determined based on the number of states of each sensor, which is determined based on the regression coefficient of each sensor. When the absolute value of the regression coefficient of a sensor is higher, the number of states of the sensor is increased.

We use a variation property, shown in Figure 5, as the number of states based on the regression coefficient. When the regression coefficient of a sensor is less than m_β , the minimum number of states is 1; when it is greater than m_α , the maximum number of states is v_i^* ; and when it is from m_β to m_α , the number of state increases gradually. The parameters m_β and m_α are determined by a human. The formulation of the property is determined by (10). v_i is the number of states of sensor i :

$$v_i = \begin{cases} 1 & (|a_i| < m_\alpha), \\ \left\lceil \frac{v_i^* - 1}{m_\beta - m_\alpha} |a_i| + \frac{m_\beta - m_\alpha v_i^*}{m_\beta - m_\alpha} \right\rceil & (m_\alpha \leq |a_i| \leq m_\beta), \\ v_i^* & (m_\beta < |a_i|). \end{cases} \tag{10}$$

Here, v_i^* is determined based on the performances of the sensor. We focus on the resolution and maximum range as the performance of the sensor. When the performance of the resolution and maximum range of a sensor are high, the robot can describe more states. v_i^* is the state number, which can be calculated based on the resolution and maximum range of the sensor, as in (11), where $g_{\max,i}$ is the maximum range of sensor

FIGURE 6: Example of merged state of a sensor.

i , $g_{\min,i}$ is the minimum range of sensor i , b_i is resolution of sensor i , and v_i^* is calculated for each sensor. We define a state of a sensor on v_i^* as a “state unit”:

$$v_i^* = \frac{g_{\max,i} - g_{\min,i}}{b_i}. \quad (11)$$

3.4. Construction of a Temporary Q-Space for Action Selection. The robot constructs a temporary Q-space based on v_i . The temporary Q-space consists of a unit, which is several state units merged together. We show an example of a merged state space in Figure 6. In this example, v_i^* is 9 and the number of state units is 9. When v_i is 3, the temporary Q-space is constructed of three states obtained by merging three state units. This example is focused on one sensor. All the installed sensors are merged according to v_i .

Here, when the number of states is v_i , the number of state units in a merged state v_i' is calculated by the following equation:

$$v_i' = \left\lceil \frac{v_i^*}{v_i} \right\rceil. \quad (12)$$

The Q-value of each state unit in merged targets is averaged, as shown in Figure 7, where an example of the temporary Q-space when v_1^* and v_2^* are 6, v_1' is 3, and v_2' is 3 is depicted. For sensor S_1 , each of the states is three merged state units. For sensor S_2 , each of the states is two merged state units. Merged state units are averaged Q-values.

The robot selects an action based on the temporary Q-space for the current state. It recognizes the current state as state unit s . When the current state is s , the merged target group of the state unit is $C_i = \{v_{i,o}, v_{i,p}, \dots, v_{i,q}\}$, where $v_{i,o}, v_{i,p}, \dots, v_{i,q}$ in the group are the state units of a sensor. Similarly, the target group of sensor t is $C_t = \{v_{t,o}, v_{t,p}, \dots, v_{t,q}\}$, and the target group of sensor u is $C_u = \{v_{u,o}, v_{u,p}, \dots, v_{u,q}\}$. Here, the Q-value of the merged target in Q-space $Q(s_m, i)$ is defined by (13), where $R(s_m, a)$ is the total reward of merged state units, defined as (14), where $R(s_w, a)$ is the total reward at state $s_w = \{v_{i,w}, v_{t,w}, \dots, v_{u,w}\}$ and action a and $E(s_m, a)$ is the total number of the experiences of the state action pair (s_m, a) , defined as (15). Merging is performed for all the actions in state s_m :

$$Q(s_m, a) = \frac{R(s_m, a)}{E(s_m, a)}, \quad (13)$$

$$R(s_m, a) = \sum_{v_{i,w} \in C_i} \sum_{v_{t,w} \in C_t} \dots \sum_{v_{u,w} \in C_u} (Q(s_w, a) \cdot E(s_w, a)), \quad (14)$$

$$E(s_m, a) = \sum_{v_{i,w} \in C_i} \sum_{v_{t,w} \in C_t} \dots \sum_{v_{u,w} \in C_u} E(s_w, a). \quad (15)$$

3.5. Action Selection. The robot selects an action based on the temporary Q-space. We apply the ϵ -greedy method for action selection. This method selects the action that has the highest Q-value in the current state unit s . However, the method selects an action randomly with probability ϵ .

3.6. Action Evaluation. We apply the weighted averaging method as the action evaluation method. This method evaluates actions by assigning a weight to each reward recently obtained by the robot. When the current state unit of the robot is s and the selected action is a , Q-value ($Q(s, a)$) is updated by (16). α is a step size parameter ($0 \leq \alpha \leq 1$):

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r - Q(s, a)]. \quad (16)$$

4. Experiment to Confirm the Effectiveness of the Proposed System

4.1. Outline of the Experiment. In this section, we describe our evaluation of the effectiveness of the proposed system via an experimental robot. The experimental environment is shown in Figure 8. This environment is surrounded by walls of length 1100 mm. We prepared an experimental robot, as shown in Figure 9. The robot has two distance sensors, which measure the distance between the current position of the robot and the walls. It can recognize the current state of a sensor as the state value u , as shown in Figure 10. Its sensors are divided into 11 states every 70 mm, and each state is given a state value. In this experiment, each sensor had 11 states and the total number of states was 121. The robot has omniwheels and can move in the forward, back, left, right, and each diagonal direction but cannot turn. It can move 70 mm in one action when it is not moving in a diagonal direction. When the robot moves in a diagonal direction, it moves in the cross direction and then in the lengthwise direction.

In the experiment, the task of the robot was to move close to wall A. The robot could obtain rewards according to its distance only from wall A. It was placed at the lower right corner of the environment and when it reached wall A, an episode was considered. The experiment was concluded at n_e episodes.

This task seems simple at first glance. However, it is difficult for an RL robot. In this task, when a robot strays into an unexperienced area, its action selection becomes random, because the Q-values of the states in the area are at their initial value. Therefore, the robot strays more by repeating a random action selection and takes time to estimate the Q-values of the states and to leave the area. In addition, in this experiment, the task was being performed by an experimental robot. Therefore, it is possible that the sensor noise affected its learning of, for example, gap state recognition.

In real world use scenes, a robot is rarely required to move in a maze-like environment with many obstacles. Most use scenes are open spaces, such as a warehouse or park. Therefore, the environment we adopted is appropriate for this experiment.

We confirmed the effectiveness of the proposed system by comparing it with a conventional RL. Therefore, we prepared two types of robot for comparison, to one of which the

FIGURE 7: Example of projection for a temporary Q-space.

FIGURE 8: Experimental environment.

FIGURE 9: Experimental robot.

proposed system was applied, and to the other RL. The number of states of each sensor of the robot to which RL was applied was v^* . We compared these agents in terms of the total reward obtained after n_e episodes were completed.

TABLE 1: Experimental settings.

n_e	100
α_{ave}	0.1
ϵ	0.1
α	0.1
Initial value of Q-value	0
Maximum range of sensors g_{max}	1100 mm
Minimum range of sensors g_{min}	0
Resolution of sensors b	100 mm
m_α	0.2
m_β	0.8
The number of initial states	11
Initial importance	1.0

4.2. Experimental Setup. In this section, we explain the reward for each task and discuss the parameter settings. In task A, the robot can obtain a higher reward by increasing the distance between its current position and wall A, as defined in (17), where d_A is the state value shown in Figure 10. d_A is determined based on the actual measurement value of the distance between the current position and wall A:

$$r = 11 - d_A. \quad (17)$$

A list of the parameter settings of this experiment is given in Table 1. In this experiment, the settings of the maximum range, minimum range, and resolution of sensors remained the same. v^* was 11 according to (11). When the robot started a new episode, the Q-space and state knowledge list from previous episodes were adopted.

4.3. Experimental Results. The experimental results are shown in Figures 11–15. Figure 11 shows the importance of the sensors for the final action in each episode. The importance of the sensors is represented by the regression coefficients. In first episode, the regression coefficients of each sensor are converged in the early phase of learning. The regression of the coefficient of sensor A is greater than the threshold $m_\beta = 0.8$. On the other hand, the regression of the coefficient of sensor

FIGURE 10: State recognition of the robot.

FIGURE 11: Importance of sensors in each episode.

FIGURE 12: The number of states in each episode.

B is smaller than the threshold $m_\alpha = 0.2$. In this task, only wall A is related to reward and its importance is high. It is valid that the regression of the coefficient of the wall A sensor is high and that of wall B is low.

Figure 12 shows the number of the states of the sensors in the final action in each episode. The number of states of sensor A is the maximum, 11. The number of states of sensor B is the minimum, 1. Using these results, the robot can construct a correct temporary Q -space.

Figures 13 and 14 show the importance of the sensors and the number of states in each action in the first episode, respectively. Until the 30th action, the importance of the sensors is unstable. The reason is that the robot has insufficient knowledge to calculate the regression of the coefficient correctly. After the 31st action, the robot has sufficient knowledge and can therefore calculate the regression of the coefficient correctly. Until the 30th action, because of the instability in the number of the states of the sensors, the

FIGURE 13: Importance of sensors in each action in episode 1.

FIGURE 14: The number of states in each action in episode 1.

robot's ability to calculate the importance of the sensors is negatively affected by its insufficient knowledge.

Figure 15 shows the total number of actions in each episode. The robot using the proposed method achieves a high convergence of the number of actions as compared to that using RL. This is because the robot using RL strays into an area for which it has no experience and takes time to leave it. On the other hand, this does not occur when the proposed method is used. This is because, when the robot strays into an area for which it has no experience, it can use other Q-values learned in other states based on the constructed temporary Q-space, which consists of only important sensors. Therefore, the robot does not need to take time for learning in the unexperienced area. Thus, the robot focuses on only the important sensors and selects the suitable action by using this Q-space. These results show that the proposed method is effective for learning.

5. Conclusion

In this paper, we proposed a method in which a robot selects an action by using a temporary Q-space based on the importance of its sensors. This method assumes that there is a correlation between the sensor value and reward. The robot calculates the regression coefficient using a multiple regression equation of the sensor value and reward. The robot determines the importance of its sensors according to the regression coefficient. The higher the level of importance, the larger the number of states. To select an action, the robot constructs a temporary Q-space based on the importance of the sensors. It then selects actions based on the temporary Q-space. Thus, the robot is able to learn faster.

We examined the effectiveness of the proposed system using an experimental robot. We investigated a task for the execution of which only one sensor of a sensor pair was important. We compared the proposed system with

FIGURE 15: The number of total actions at each episode.

conventional RL. The robot used sensors whose number of states is common in the case of conventional RL.

The results showed that using the proposed system the robot could calculate the importance of the sensors correctly. In addition, convergence speed was faster than that in conventional RL. Thus, we confirmed the effectiveness of the constructed system and the proposed method.

In future studies, first, we will examine the effectiveness of the proposed system by comparing it with other autonomous state construction systems. In this study, we examined only normal reinforcement learning. It is necessary to examine the proposed system by comparing it with those proposed in related studies. Then, we will modify the proposed method. Currently, the proposed method cannot be applied in a delayed reward task, because the regression coefficient is calculated using the immediate reward in each state. When a task is a delayed reward task, the robot cannot calculate the importance of its sensors. Therefore, we will modify the proposed method so that it can be applied in a delayed reward task by using information that does not include the reward.

References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [2] T. Kondo and K. Ito, "A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 111–124, 2004.
- [3] K. J. Person, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2650–2655, 2010.
- [4] N. Navarro, C. Weber, and S. Wermter, "Real-world reinforcement learning for autonomous humanoid robot charging in a home environment," in *Towards Autonomous Robotic Systems*,

- vol. 6856 of *Lecture Notes in Computer Science*, pp. 231–240, Springer, 2011.
- [5] M. Tan, “Multi-agent reinforcement learning: independent vs. cooperative agents,” in *Proceedings of the 10th International Conference on Machine Learning*, 1993.
 - [6] M. N. Ahmadabadi, M. Asadpur, S. H. Khodaabakhsh, and E. Nakano, “Expertness measuring in cooperative learning,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’00)*, vol. 3, pp. 2261–2267, November 2000.
 - [7] M. N. Ahmadabadi and M. Asadpour, “Expertness based cooperative Q-learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 1, pp. 66–76, 2002.
 - [8] H. Iima and Y. Kuroe, “Swarm reinforcement learning algorithm based on exchanging information among agents,” *Transactions of the Society of Instrument and Control Engineers*, vol. 42, no. 11, pp. 1244–1251, 2006.
 - [9] Y. Yongming, T. Yantao, and M. Hao, “Cooperative Q learning based on blackboard architecture,” in *Proceedings of International Conference on Computational Intelligence and Security Workshops (CIS ’07)*, pp. 224–227, December 2007.
 - [10] T. Nishi, Y. Takahashi, and M. Asada, “Incremental behavior acquisition based on reliability of observed behavior recognition,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’07)*, pp. 70–75, November 2007.
 - [11] M. Asada, S. Noda, and K. Hosoda, “Action-based sensor space categorization for robot learning,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’96)*, pp. 1502–1509, November 1996.
 - [12] H. Ishiguro, R. Sato, and T. Ishida, “Robot oriented state space construction,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’96)*, pp. 1496–1501, November 1996.
 - [13] K. Samejima and T. Omori, “Adaptive internal state space construction method for reinforcement learning of a real-world agent,” *Neural Networks*, vol. 12, no. 7-8, pp. 1143–1155, 1999.
 - [14] A. J. Smith, “Applications of the self-organising map to reinforcement learning,” *Neural Networks*, vol. 15, no. 8-9, pp. 1107–1124, 2002.
 - [15] K. T. Aung and T. Fuchda, “A proposition of adaptive state space partition in reinforcement learning with voronoi tessellation,” in *Proceedings of the 17th International Symposium on Artificial Life and Robotics*, pp. 638–641, 2012.
 - [16] Y. Takahashi, M. Asada, and K. Hosoda, “Reasonable performance in less learning time by real robot based on incremental state space segmentation,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96)*, pp. 1518–1524, November 1996.