Autonomous Environment Recognition by Robotic Manipulators

Proceedings of 2001 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 29 - August 1, 2001, Banff, Alberta, Canada

Autonomous Environment Recognition by Robotic Manipulators

Kei SENDA*
Graduate School of Engineering
Osaka Prefecture University
senda@aero.osakafu-u.ac.jp

Yuzo OKANO
Graduate School of Engineering
Osaka Prefecture University

Abstract

This paper discusses methods of autonomous environment recognition and action by a robotic manipulator working with dynamic interaction to the environment, e.g., assembling. A method automatically recognizes the contacting situation with the work site from the sensor outputs and the robotic manipulator motion. The autonomous recognition then discriminates the constraint conditions at manipulator hand using the self-organizing map that is a kind of unsupervised learning of neural networks. The discrimination of the constraint conditions is successfully demonstrated by a numerical simulation of a 3-link SCARA type manipulator. Another is for the cognitive action. Some approaches based on the reinforcement learning are proposed. They give models of cognitive actions and approaches to so-called frame problem obstructing efficient learning and action.

1 Introduction

Robots achieve limited tasks in only well-organized worksites. In order to extend robot applications, we must extend the task-accomplishment ability in various environments . Hence, cognitive actions are important by a robotic manipulator having dynamic interaction to the environment, e.g., assembling.

In peg-in-hole operation, for example, a man can easily insert a peg in a hole after a trial-and-error procedure. The man must find out a contacting situation of the peg to the holes adaptively and act suitably. In the same way, a robot can automatically adapt various environments if it recognizes and acts when the work environments are different from the expected in advance.

There are many studies relating to snoring, e.g., reference[1]. But, most of them are not for autonomous recognition by robots. Because of the variation of the environments, the robotic autonomous recognition is very important. For the purpose, this paper proposes some methods for environment recognition.

This paper firstly proposes a method of automatic discrimination of contact situations from the sensor outputs and the robotic manipulator motion by using self-organization of a neural network. In addition, some approaches based on the reinforcement learning are proposed for the autonomous recognition and action. They are approaches to so-called frame problem obstructing efficient learning and action.

*Acknowledgement: A part of this work is financially supported by a Grant-in-Aid for Scientific Research from Ministry of Education, Science, Culture and Sports of Japan and a Grant-in-Aid for advanced scientific research from Osaka Prefecture University.

2 Model of Constrained Manipulator

By using Lagrange undetermined multipliers[2], the equation of motion of a robot manipulator with the geometric endpoint constraint illustrated in Fig. 1 is given as[3, 4]

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) + \frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{\theta}}\boldsymbol{\lambda} = \boldsymbol{\tau} \quad (1)$$

$$\boldsymbol{\phi}(\boldsymbol{y}) = [\phi_1(\boldsymbol{y}), \cdots, \phi_m(\boldsymbol{y})]^T = \boldsymbol{0} \quad (2)$$

where $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_n]^T$ denotes the joint angles, $\boldsymbol{M}$ the inertia matrix, $\boldsymbol{h}$ the centrifugal and Coriolis forces, $\boldsymbol{g}$ the gravity forces, $\boldsymbol{\tau}$ the joint torques, $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda]^T$ the Lagrange undetermined multipliers, $\boldsymbol{\phi}$ the constraint equations, and $\boldsymbol{y}$ the task coordinates. Eq. (1) is in the case that friction forces do not act between the constraint surface and the endpoint of the manipulator. The friction forces can be regarded as the external forces affecting at the end-effector if necessary.

The coefficient matrix of $\boldsymbol{\lambda}$ in Eq. (1) is

$$\frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{y}^T} \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}^T} \right)^T = \boldsymbol{J}^T \frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{y}} \quad (3)$$

where $\boldsymbol{J}$ denotes Jacobian matrix.

Lagrange undetermined multiplier λ_i stands for the constraint force corresponding to ϕ_i that has a force or a torque dimension expressed in the constraint coordinate (Fig. 2).

In case of the 3-link SCARA manipulator moving in the xy-plane Fig. 1, manipulation variables and constraint equations are given as

$$\boldsymbol{y} = [x, y, \alpha]^T \quad (4)$$

$$\phi_1 = x + y + b = 0 \quad (5)$$

$$\phi_2 = \alpha + c = 0 \quad (6)$$

$$\phi_3 = x - y + d = 0 \quad (7)$$

where $\alpha = \theta_1 + \theta_2 + \theta_3$ denotes the attitude angle of the end-effector, and b, c, d are constants.

3 Environment Classification by Using SOM

3.1 Problem Definition

The 3-link SCARA manipulator with a force torque sensor in the wrist is constrained on the smooth constraint surface by the peg (Fig. 1). Here constraint 0 or 1 or 2 or 3 (Fig. 2) appears with constraint equations. If ϕ_i's are lineally independent and we identify constrain equation ϕ_i does not work when $\lambda_i = 0$, the number of effective constraint equations is regarded as the number of non-zero λ_i.

This paper discusses a problem to discriminate the constraint conditions when the robot manipulator does

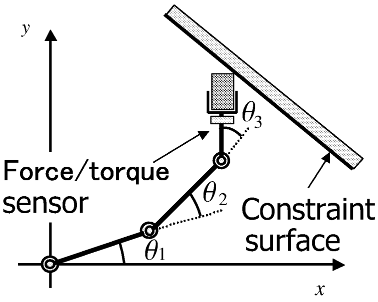

Figure 1: Manipulator constrained at the endpoint

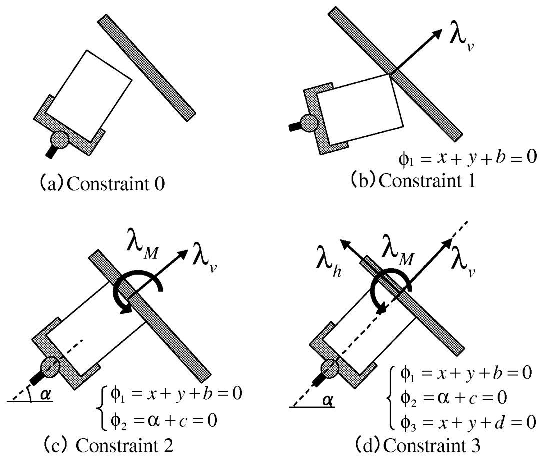

Figure 2: Constraint situations, ϕ_i's and λ's

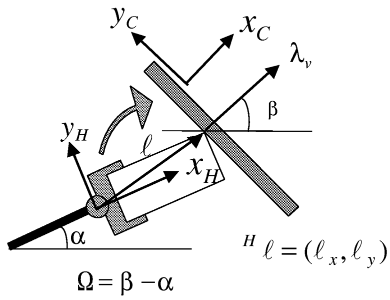

Figure 3: Transformation to constraint coordinates

not have any information about constraints in advance. In other words, the problem is to guess the combination of acting ϕ_i's in the unknown constraint.

3.2 Relations Between Sensor Information and Constraint Forces

In a real system, we cannot obtain the multipliers $\boldsymbol{\lambda}$ but the force information $\boldsymbol{\xi} = [\xi_x, \xi_y, \eta_z]^T$ from the force/torque sensor, where ξ_x and ξ_y are the x and y-directions forces expressed in Σ_H and η_z is the torque in z-axis. In the following discussion, we derive a transformation from $\boldsymbol{\xi}$ to $\boldsymbol{\lambda}$ in the case of Fig. 1. The superscripts at the left shoulder $^H(\ )$ and $^C(\ )$ represent coordinate frames. The Σ_H denotes the hand coordinate frame fixed at the wrist of the manipulator. The Σ_C is the constraint coordinate frame, whose origin is in the contact point, x-axis directs to the normal line of the constraint surface, and y-axis directs to the tangential line. Consider the case that neither the constraint equations nor the shape of the peg are given, and we discuss if $^H\boldsymbol{\xi}$ can always be transformed to $^C\boldsymbol{\lambda}$ (Fig.3). The following equation represents the transformation from $^H\boldsymbol{\xi}$ to $^C\boldsymbol{\lambda}$

$$\begin{bmatrix} C_\Omega & S_\Omega & 0 \\ -S_\Omega & C_\Omega & 0 \\ \ell_y & -\ell_x & 1 \end{bmatrix} \begin{bmatrix} \xi_x \\ \xi_y \\ \eta_z \end{bmatrix} = \begin{bmatrix} \lambda_v \\ \lambda_h \\ \lambda_M \end{bmatrix} \tag{8}$$

where $S_\Omega = \sin\Omega$, $C_\Omega = \cos\Omega$, and

Ω : angle between x-axis of Σ_H and x-axis of Σ_C

ℓ_x, ℓ_y : x, y-positions of the contact point expressed in Σ_H

Eq. (8) gives in the following form:

$$\xi_x C_\Omega + \xi_y S_\Omega = \lambda_v \tag{9}$$
$$-\xi_x S_\Omega + \xi_y C_\Omega = \lambda_h \tag{10}$$
$$\xi_x \ell_y - \xi_y \ell_x + \eta_z = \lambda_M \tag{11}$$

In the case that ℓ_x, ℓ_y, and Ω are unknown, the method of transformation from $^H\boldsymbol{\xi}$ to $^C\boldsymbol{\lambda}$ is shown below. Constraint states in Figs. 2 (a) to (d) are called constraints 0 to 3, respectively. The transformation for constraint 0 is not considered because no constraint force acts in the case.

Constraint 1 Because constraint 1 is realized when $\lambda_h = \lambda_M = 0$, Eq. (11) becomes

$$\xi_x \ell_y - \xi_y \ell_x + \eta_z = 0 \tag{12}$$

If the motion of constraint 1 appears for a time period ($t_1 \sim t_n$), there exist ℓ_x and ℓ_y satisfying

$$\left\{ \begin{array}{rcl} \eta_z(t_1) & = & \xi_y(t_1)\ell_x - \xi_x(t_1)\ell_y \\ \eta_z(t_2) & = & \xi_y(t_2)\ell_x - \xi_x(t_2)\ell_y \\ & \vdots & \\ \eta_z(t_n) & = & \xi_y(t_n)\ell_x - \xi_x(t_n)\ell_y \end{array} \right. \tag{13}$$

All equals may not hold simultaneously because of sensor noise and so on. Hence ℓ_x and ℓ_y are determined by the least-square-error method. At the same time, angle β between the constraint force and the horizon is obtained by considering $\dot{\boldsymbol{\phi}} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{y}^T}\dot{\boldsymbol{y}} = \mathbf{0}$. The Ω is also given by $\Omega = \beta - \alpha$. Hence $\boldsymbol{\lambda}$ can be obtained by Eqs. (9) to (11).

Constraint 2 The conditions realizing constraint state 2 are that $\lambda_h = 0$ holds and ℓ_x, ℓ_y, and Ω are constant. Considering $\lambda_h = 0$, Ω is obtained by Eq. (10). In constraint 2, the ratio of ξ_x to ξ_y is constant because Ω is unvaried. Because $(\xi_x(t_i), \xi_y(t_i))$'s in Eq. (13) are not independent, ℓ_x and ℓ_y cannot be determined. This means that the constrained equation $\phi_1(\ell_x, \ell_y) = 0$ is also represented by $\phi_1'(\ell_x', \ell_y') = 0$. Hence $^C\boldsymbol{\lambda}$ can be calculated by Eqs. (9) to (11) using ℓ_x and ℓ_y in the designed values.

Constraint 3 The conditions realizing constraint 3 are ℓ_x, ℓ_y and Ω are constant. In the same way of constraint 2, ℓ_x, ℓ_y and Ω cannot be determined. The constraint force $\boldsymbol{\lambda}$ can be obtained by Eqs. (9) to (11) using ℓ_x, ℓ_y, and Ω in the designed values.

Hence $^H\boldsymbol{\xi}$ can be always transformed to $^C\boldsymbol{\lambda}$.

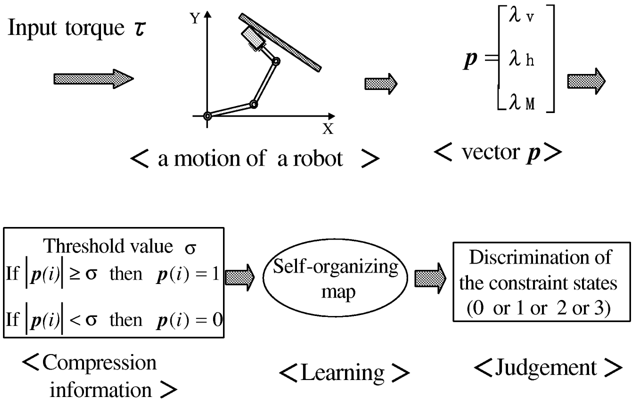

Figure 4: Outline of proposed method

3.3 Constraint State Calcification

Outline

The proposed method is outlined in Fig. 4. When the input torque is given to the manipulator represented by Eq. (1), the sensor information about the joint angle and the wrist force is gotten during the motion of the manipulator. The sensor information is processed appropriately and put into a self-organizing map[5, 6].

In the self-organizing map, input patterns are discriminated, and a weighting matrix $\boldsymbol{W}$ is made as described later. By using the obtained weighting matrix, the constraint states can be discriminated correctly.

The sensor information processing is necessary because it is almost impossible to discriminate the constraint states correctly if the raw sensor information is inputted to the self-organizing map. Hence the processed information, e.g., the quantized sensor signal in discrete-time, is used in a later section.

Maximum detector using competing system

Firstly we introduce a maximum detector used in the self-organizing map. Dynamics of the basic competing system with feedback type lateral inhibition is represented as follows, which detects the maximum[7]:

$$
\begin{aligned}
\frac{du_i(t)}{dt} &= -u_i + c_1 y_i - c_2 \sum y_i + s_i \qquad (14)\\
y_i &= 1(u_i), \qquad i = 1, 2, \cdots, M \qquad (15)
\end{aligned}
$$

where M denotes the number of the network unit, s_i the input to unit i, and u_i the internal state of unit i. Each unit outputs y_i is the step function $1(u_i)$ of Eq. (15). The output y_i of unit i is fedback to itself by a connection weight $c_1 > 0$, and to other units by a connecting weight $c_2 < 0$. All units have positive values as the internal states. The parameter c_1 and c_2 satisfied the conditions

$$
c_2 + 1 < c_1 < 2c_2 - 1 \qquad (16)
$$

Each unit is trying to decrease other units' internal states by the lateral inhibition. The internal state of the unit with the maximum input stays positive for the longest. The others become negative and they output zeros. After this, other units do not inhibit the last positive unit, and the unit keeps firing. Hence only the unit with the maximum input generates 1 and this network acts as the maximum detector.

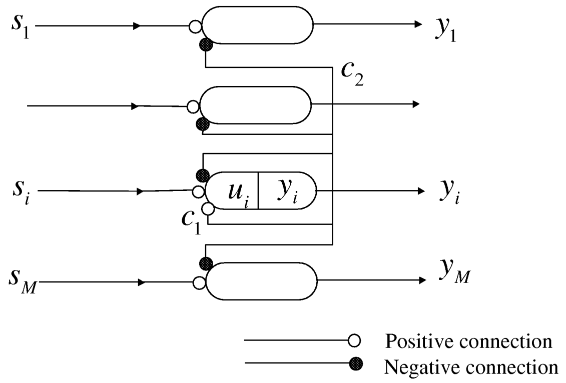

Figure 5: Maximum detector using basic competing system

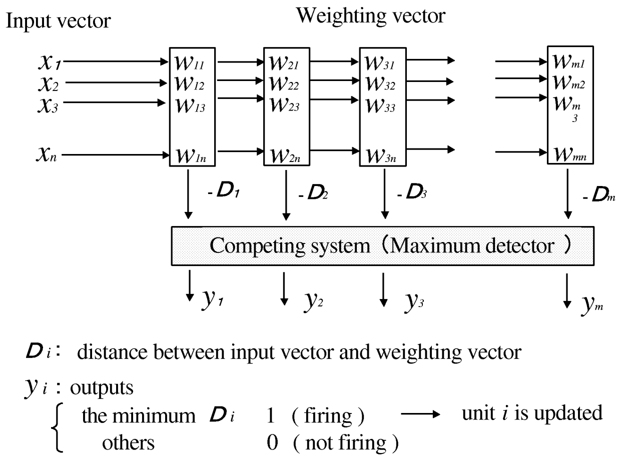

Figure 6: Self-organizing map

Self-organizing map

For the constraint states discrimination, used here is the self-organizing map (SOM), which is a kind of unsupervised learning of neural networks[5, 6, 7]. The neural network automatically acquires an ability to classify the inputs into some groups according to their similarity. As illustrated in Fig. 6, the input vector $\boldsymbol{x} = [x_1, \cdots, x_n]^T$ is multiplied by weighting vector $\boldsymbol{w}_i = [w_{i1}, \cdots, w_{im}]^T$ and input to unit i, where w_{ij} denotes connecting weight between unit i and input x_j. Represented by C is the unit whose weighting vector is most similar to the input vector. Output y_i is given as

$$
\begin{aligned}
y_i &= 1 \quad \text{for } i = C\\
y_i &= 0 \quad \text{for } i \neq C
\end{aligned} \qquad (17)
$$

When we detect unit C, the Euclidian norm $||\boldsymbol{x} - \boldsymbol{w}||$ between the input vector and the weighting vector is calculated. By inputting $-||\boldsymbol{x} - \boldsymbol{w}||$ into the above maximum detector, the weighting vector nearest to the input vector is detected, where the nearest vector is most similar to the input vector. The learning is proceeded as

$$
\begin{aligned}
\boldsymbol{w}_i(t+1) &= \boldsymbol{w}_i(t) + \eta(\boldsymbol{x}(t) - \boldsymbol{w}_i(t))\\
&\quad \text{for } i = C\\
\boldsymbol{w}_i(t+1) &= \boldsymbol{w}_i(t) \quad \text{for } i \neq C
\end{aligned} \qquad (18)
$$

where η is a learning rate.

3.4 Numerical Simulations

Simulation conditions

Appropriate torque $\boldsymbol{\tau}$ is inputted to the manipulator of Fig. 1. Constraint 0 appears for a period of discrete time [0 20], constraint 1 for [21 40], constraint 2 for [41 60], constraint 1 for [61 80], and constraint 2 for [81 100], constraint 3 for [101 120]. By using sensor information of the periods, differences of the constraint states are learned. After that, the manipulator moves in the same way again, we demonstrate if the robot can discriminate the constraint states by numerical simulation. The sensor information is processed in the two ways mentioned later. Impulsive forces are not contained in the numerical simulation that might be generated at the constraint switching.

A method using quantized vector

Introduce $\boldsymbol{p}$ of $\boldsymbol{\lambda}(t)$ as

$$\boldsymbol{p} = [1 \ \lambda_v \ \lambda_h \ \lambda_M]^T \tag{19}$$

where the first element of $\boldsymbol{p}$ is a dummy parameter that is introduced not to be the zero vector. When the vector $\boldsymbol{p}$'s are directly inputted to the self-organizing map, constraint states are not classified correctly. In this paper, $\boldsymbol{p}$'s are quantized for the information compression. The quantized vector $\boldsymbol{p}_{vq}$ is given as

$$\boldsymbol{p}_{vq}(i) = \begin{cases} 1 & \text{for} \ |\boldsymbol{p}(i)| \geq \sigma \\ 0 & \text{for} \ |\boldsymbol{p}(i)| < \sigma \end{cases} \tag{20}$$

where $\sigma(\ll 1)$ denotes a threshold value and $\boldsymbol{p}(i)$ denotes the i-th element of $\boldsymbol{p}$. The $\boldsymbol{p}_{vq}$ is composed either 0 or 1. The $\boldsymbol{p}_{vq}$ of each constraint state is represented as

$$\begin{aligned} \text{Constraint 0} &: [1 \ 0 \ 0 \ 0]^T \\ \text{Constraint 1} &: [1 \ 1 \ 0 \ 0]^T \\ \text{Constraint 2} &: [1 \ 1 \ 0 \ 1]^T \\ \text{Constraint 3} &: [1 \ 1 \ 1 \ 1]^T \end{aligned}$$

The weighting matrix $\boldsymbol{W}$ has $\boldsymbol{p}_{vq}^T$ of each constraint state in each row vector after learning by the use of $\boldsymbol{p}_{vq}$'s:

$$\boldsymbol{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{21}$$

The robot manipulator discriminates the constraint states by using Euclidian norm to measure the similarity of vectors.

The sensor information $\boldsymbol{p}$ is inputted into the self-organizing map after the quantizing. The outputs of the self-organizing map are labeled 0 to 3, which is corresponding to the label of the fired unit. Fig. 7 shows that the robot manipulator successfully classifies the constraint states.

In this method using quantized vector, constraint force $\boldsymbol{\lambda}$ is expressed in the constraint coordinate frame Σ_C whose bases are the column vectors of the matrix $\frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{y}}$. It is regarded that we select the convenient coordinate to classify successfully. By quantizing the vector $\boldsymbol{p}$ with the elements of 0 or 1, each constraint state is identified as one of the quantized vectors. As a result, each row vector in $\boldsymbol{W}$ one-to-one corresponds to the constraint state (Fig.8) and the discrimination is successful.

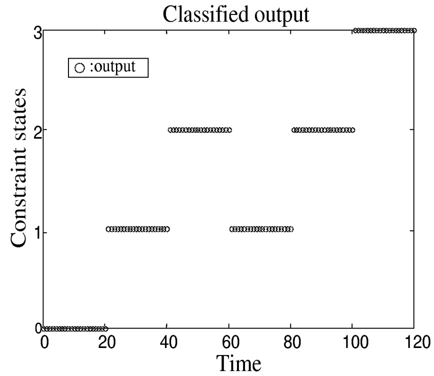

Figure 7: Constraint state classification by SOM

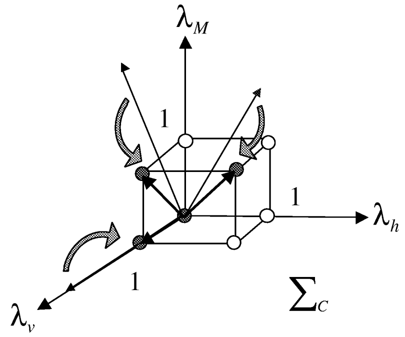

Figure 8: Mapping of the quantized input vector

A method using distance from state sets

Sets of $\boldsymbol{\lambda}$ in the constraint states are illustrated in Fig. 9. It is easily understood that the set of constraint 0 is the origin, constraint 1 the λ_v axis, constraint 2 the $\lambda_v - \lambda_M$ plane, constraint 3 the rest of the $\boldsymbol{\lambda}$ space. The distances from the sets are

$$d_1 : \text{the distance from the origin} \tag{22}$$
$$d_2 : \text{the distance from the } \lambda_v \text{axis} \tag{23}$$
$$d_3 : \text{the distance from the } \lambda_v\text{-}\lambda_M \text{ plane} \tag{24}$$

Then define the vector $\boldsymbol{d}_i = [d_1, d_2, d_3]^T$. The $\boldsymbol{d}_i$ is represented as follows for each constraint:

$$\boldsymbol{d}_0 = [0 \ , \ 0, \ 0]^T \tag{25}$$
$$\boldsymbol{d}_1 = [d_1 \ , \ 0, \ 0]^T \tag{26}$$
$$\boldsymbol{d}_2 = [d_1 \ , \ d_2, \ 0]^T \tag{27}$$
$$\boldsymbol{d}_3 = [d_1 \ , \ d_2, \ d_3]^T \tag{28}$$

The constraint states are also classified by using the vector $\boldsymbol{d}_i$ if the above quantized vector method is applied. The result is omitted because it is similar to the previous result.

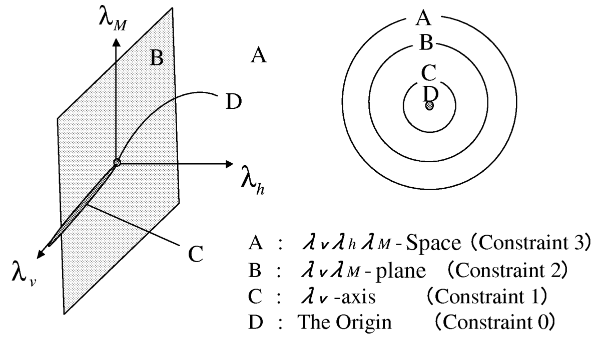

Figure 9: Inclusion relation of the constraint states

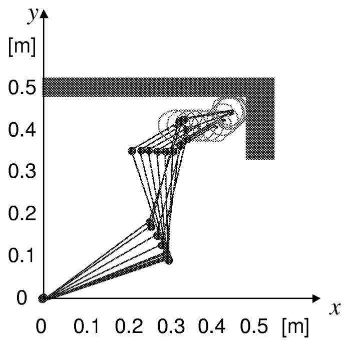

Figure 10: Motion Acquired by Q-Learning

4 Cognitive Action

4.1 Models of Cognitive Actions

Investigations of skilled human operators point out a change of "observation". At the beginning, the operators must recognize, plan, choose from actions, etc. and difficult to work quickly. As the persons repeat working, they skip the internal processes relating the environment recognition with much effort, and their environmental observation change to indicate efficient and right action. This can be considered that a knowledge-based behavior changes to a rule-based or skill-based behavior in Rasmussen's model and amount of the information process reduces[8]. The efficient observation is similar to feature-based action[9]. It is called co-provision with a dual-loop feedback structure that the environmental observation provides and organizes behavior and the resultant behavior provides observation again[10]. In the following sections, the change of observation and the co-provision are modeled as the selection of state variables, the categorization of state, and the use of categorized state space. They are also approaches to the frame problem[11] using recognition.

4.2 Formulation based on Reinforcement Learning

As shown in Fig. 10, considered here is a task where the 3-link SCARA type manipulator places the component and presses it against the corner of walls in desired direction and force to assemble.

This is simplified from the Peg-in-Hole task and no friction is contained for simplicity. Visual information is not used on the grounds that the robot uses only the forces at hand and joint angles in the final assembling because vision measurement error is not ignored.

As a solution for the reinforcement-learning problem, Q-learning[12] is employed Its formulation is base on a finite discrete space, where time, state, and action are discretized as well as general Q-learning. The system state is defined as follows. Convergence of learning is guaranteed only if the system's state space is constructed so as to determine its future state relating the task form current state and action. Hence it is reasonable to use the state variables of the equations of motion (1) of the robot manipulator with the geometric endpoint constraint as:

$$[x, y, \alpha, \dot{x}, \dot{y}, \dot{\alpha}, f_x, f_y, n_z] \tag{29}$$

where they are x, y positions and direction of hand and component, their velocities, and applied forces in x, y, α directions from the environment, respectively. They all are measured form the sensors.

The following controller is used to generate actions:

$$\boldsymbol{\tau} = -\boldsymbol{J}^T\boldsymbol{K}_P(\boldsymbol{y} - \boldsymbol{y}_r) - \boldsymbol{K}_D\dot{\boldsymbol{\theta}} \tag{30}$$

where $\boldsymbol{\tau}$ is control input to the manipulator, $\boldsymbol{J}$ Jacobian matrix, $\boldsymbol{y}$ manipulation variable vector, $\boldsymbol{y}_r$ reference of $\boldsymbol{y}$, $\boldsymbol{\theta}$ joint variable vector, $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ feedback gain matrices, respectively. For the reference manipulation variable $\boldsymbol{y}_r^{(i)}$ at time i, $\boldsymbol{y}_r^{(i+1)}$ is given by $\boldsymbol{y}_r^{(i+1)} = \boldsymbol{y}_r^{(i)} + \delta\boldsymbol{y}_r^{(i)}$. Action at time i is considered as the $\delta\boldsymbol{y}_r^{(i)}$. The robot regards as the task having been achieved at a target state, where reword is given. In the target state, all velocities and n_z are zeros, and α, f_x, and f_y are specified values.

In this example, number of state is a few millions because of many degree of the state space. The learning has not been converged in 50 hours by using Pentium III 500 MHz/Matlab since much time is consumed for numerical simulations as well as the may states. One must reduce the states from a point of view of the recognition.

4.3 Change of Observation and State Space

One may wonder if all state variable in Eq. (29) are really needed. The learning has been converged using

$$[x, y, \alpha, f_x, f_y] \tag{31}$$

as state variables. One of the obtained optimal behavior is illustrated in Fig. 10. The optimal behavior is achieved from any initial state after the learning is converged. The state variables are reduced because some of the state variables in Eq. (29) are not necessary for the task achievement and the sampling time for learning is longer than the settled time of the control (30). The learning has been converged in 4 hours because the number of states has been reduced by 1/1000. An algorithm with a decision tree is used to find the state variables in Eq. (31). It takes 5 hours for the convergence of learning including this state variables finding algorithm.

This is an approach to find the minimum sufficient state space for the learning convergence as well as to ease the frame problem. This is also a model of the change of observation because the notable information in state variables is becoming clear as one is learning.

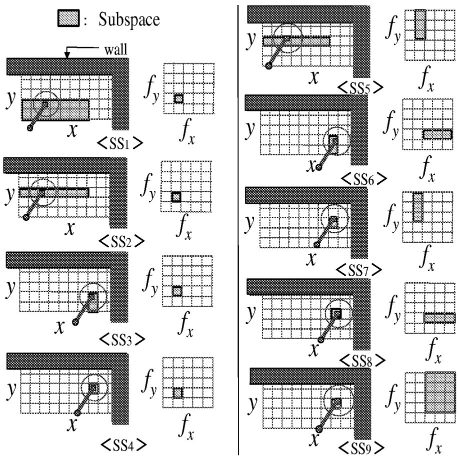

Figure 11: Categorized states

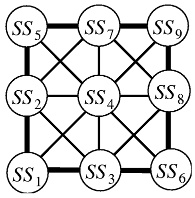

Figure 12: Connection of Sub-Spaces

4.4 Categorization and Learning with Subspaces

One can categorize the states by differences of interaction between the manipulator and the environment. There are many states, e.g., the robot move freely without constraint force, it moves freely except x direction because of constraint force f_x, etc. The states are categorized by the interaction as illustrated in Fig. 11

A state is described by a pair of two graphs of the x–y and the f_x–f_y at hand where 4 variables are used in Eq. (31). In the figure, each of SS_1 to SS_9 is a set of states belonging to each category. They are subspaces of the entire state space. For instance, the manipulator does not contact to any walls in a state of SS_1, it contacts to the upper wall in SS_5, and SS_2 is a subspace of their border. In each subspace, state transitions from any states by an action are same. Note that the categorization is dependent on the actions that one can take.

Connecting relation among the subspace is illustrated in Fig. 12. The lines between subspace shows existence of actions and the thickness indicates number of actions. The connecting relation enables to decide the action rule through a reinforcement-learning problem where each subspace is treated as a state. In the example in Fig. 10, the manipulator moves from SS_1 to SS_9. In order to achieve the determined action, the manipulator decides its action rule in each subspace through a reinforcement-learning problem whose subtask is the tangent between subspaces. The number of states can be reduced again for learning in each subspace and the learning becomes more efficient. In this example, the number of states is reduced by 1/10 form that in Eq. (31). As a result, the learning has been converged in 30 minutes.

The categorization dependent on the selectable actions can be regarded as the change of observation dependent on selectable skills. The action decision based on the subspace can be considered as the behavior organization followed by the change of observation. Moreover, if the organized behavior with the subtasks of subspace transitions becomes a skill, one can consider the rule-based behavior changes to skill-based behavior. The change from the rule-based to the skill-based may change the observation. The co-provision of observation and action can be modeled in the above. This is also an approach to ease the frame problem.

5 Concluding Remarks

This paper has proposed a method of environment recognition by a robotic manipulator working with dynamic interaction to the environment. It has been demonstrated that the robotic manipulator has discriminated the constraint conditions by the autonomous learning of neural networks when sensor information has been suitably processed, even if constraint equations were unknown.

In addition, some approaches based on the reinforcement learning have been proposed for the autonomous recognition based on the selectable actions. To find the minimum sufficient state space for the learning convergence has been modeled as the change of observation. The categorization of state and the learning with subspaces has suggested the co-provision of observation and action. They have been the approaches to the frame problem obstructing efficient learning and action.

References

[1] F. Naghdy, J. Lidbury, and J. Billingsley, "Robot Force Sensing Using Stochastic Monitoring of the Actuator Torque," *Robots and Automated Manufacture*, pp. 139–156, 1985.

[2] H. Goldstein, Classical Mechanics, Addison-Wesley Publish Company, 1980.

[3] S. Arimoto, Dynamics and Control of Robots, Asakura Publishing, Tokyo, 1990. (in Japanese)

[4] S. Arimoto, Mathematics in Systems and Controls, Iwanami Publishing, Tokyo, 1993. (in Japanese)

[5] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, Berlin, Heidelberg, 1989.

[6] T. Kohonen, Self-Organizing Map, Springer-Verlag, Berlin, Heidelberg, 1995.

[7] Y. Nhishikawa et al., Neural Networks as Applied to Measurement and Control, Asakura Publishing, Tokyo, 1995. (in Japanese)

[8] J. Rasmussen, "Skills, Rules, and Knowledge," *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, pp. 257–266, 1983.

[9] D. P. Bertsekas and J. N. Tsitsiklis: Neuro-Dynamic Programming, Athena Scientific, Belmont, 1996.

[10] T. Sawaragi, "Proficient Skills Embedded with in Human, Machine and Environment," *J. Society of Instrument and Control Engineers*, vol. 37, no. 7, pp. 471-476, 1998. (in Japanese)

[11] F. M. Brown, The Frame Problem in Artificial Intelligence, Kaufman, 1987.

[12] R. S. Sutton and A. Barto, Reinforcement Learning, MIT Press, Cambridge, 1998.