Realization of a visuo-motor system based on multiple self-organizing maps in a 3D space

Ryunosuke UEHARA† Min HAN‡ Nobuhiro OKADA†
Eiji KONDO†

†*Department of Intelligent Machinery and System*
Graduate School of Engineering, Kyushu University
744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan

‡*TOYOTA AUTO BODY CO., LTD.*
100,Kanayama Ichiriyama-cho
Kariya,Aichi 448-8666, Japan

Abstract

This paper describes a visuo-motor system which realizes a feasible collusion avoidance of a manipulator in an unknown environment. In order to handle spaces occluded by obstacles, we adapt the plural cameras system and multiple self-organizing maps (SOMs).Each self-organizing map is directly connected to the camera system and trained to perform motion control, by which the joint angle of the manipulator are determined. In our visuo-motor system, neither any priori knowledge about the manipulator nor the camera parameters are required. In addition, the system is robust to change in its geometry. Simulation result shows that the proposed learning method ensures that the manipulator moves smoothly and consistently in whole workspace even using multiple maps. In this paper, we validated the proposed approach by an experiment and confirmed that proposed method realizes collision avoidance for the visio-motor system in a 3D space. Thus, we presented simulation result that the system overcomes the collusion problem in cluttered environments and variety of obstacle shapes by increasing the number of cameras and self-organizing maps.

1 introduction

Collusion avoidance is a basic problem that a robot handles its end-effector avoiding obstacles cluttered in an environment performing the primary task for autonomous or industrial robots. To plan a path avoiding collision in an intricate environment, two contrasting approaches have been studied.

First, the high-level path planning is to find globally a collision avoidance path in the configuration space. One example of a global method is PRM (Probabilistic Roadmap Method) that computes a cell decomposition of the free space and uses a search graph based on this decomposition [1]. RRT (Rapidly-exploring Random Trees) is a roadmap method where a set of canonical paths is used to cover the components of the free space, and the planning task is reduced to determine a connection to the canonical paths [2, 3]. However, in these approaches, an exact, known and static environment model is required. In addition, the calculation time grows exponentially with the geometry complexity and the number of degree of freedom (DOF).

Therefore, local path planning techniques are potentially more efficient in robot motion planning when the environment is unknown of only partially known. An efficient local path planning method is the potential field method which has been widely used in collision avoidance [4]. In this method, a potential function is defined in the free space, based on an attraction component from the goal point and a repulsion component from the obstacle boundaries, and the planning process becomes to a determination of the global minima of the potential function using a greedy and local search.

Alternatively, obstacle avoidance can be solved online by a robot controller at the low-level, which is focused on the problem of controlling a redundant robot so that the end-effector tracks a given path in the workspace as closely as possible and simultaneously ensures that the links avoid obstacles. Reasoned as above, such techniques naturally depend on the use of different control frameworks [5, 6].

On the other hand, some methods tried to integrate a task planning and a motion control, motivated by Khatibs work [7]. This work and a few other integrated architectures [8] have utilized methods based on potential fields in their reactive control algorithms, while their planning and interface techniques differ [9]. The SOMs also can be used for path-planning or trajectory formation tasks [10]. After the mapping has been established, a path is generated from any initial position to a given target, e.g., to guide an end-effector of a robot manipulator in the presence of obstacles within the workspace. Using the TRN model [11], showed that a locally optimized path can be determined by minimizing the Euclidean distance from the current position to a given target position. However, a collision check was necessary in the path planning, and the proposed method was only investigated by using a non-redundant manipulator.

In our studies, we integrate the path planning of the end-effector and SOMs to achieve collision avoidance. The SOMs are learned to perform motion control, by which joint angles of the manipulator are determined. The learning promises to make the manipulator reach targets precisely with obstacle-free poses. The path planning system plans a collision-free path for the end-effector from an initial point to a target point in the image spaces. The proposed collision avoidance approach differs from others in: (1) The system only needs to plan a collision-free path for the end-effector; the computational cost of the path planning does not increase exponentially even for a high dimensional redundant manipulator. (2) The obstacle-free poses of the manipulator are achieved in the learning of the SOMs, so collision checking is not necessary in whole path planning process.

2 visuo-motor system

Visuo-motor system which we propose is illustrated in Fig.1. The system contains:

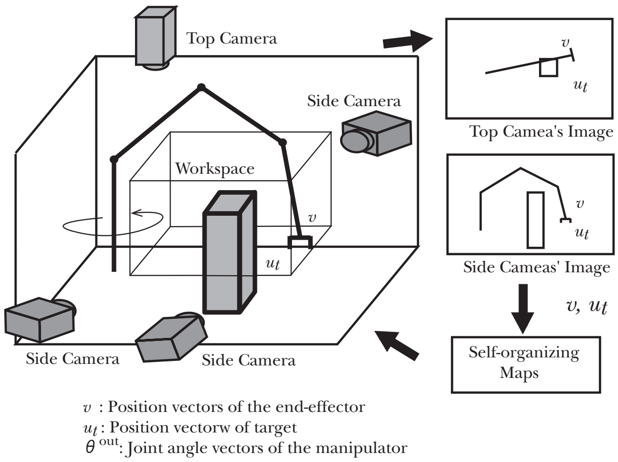

Fig. 1: Simulation model

1. A 4-degree of freedom redundant manipulator moving in a 3D space.
2. The plural CCD cameras.
3. Multiple related self-organizing maps.

CCD cameras are used to acquire information about obstacles and to recognize the position of the target, the location of the end-effector and the pose of the manipulator while learning. Based on visual information provides by cameras, each SOM learns projections that convert the image vectors of targets in the image spaces into joint angles vectors of the manipulator. The manipulator is ordered by a set of joint angle commands θ_{out} which are outputs of SOMs.

In our previous works, the system could not deal with spaces occluded by obstacles. In this paper, a redundant camera system is introduced to overcome the occlusion problem [13]. One camera observes the workspace from the top and other cameras are arranged from the sides. Because the valid workspace is increased obviously by adding the redundant camera, multiple related SOMs are employed in our system. As shown in Fig.1, the projections of a target point u_t in side cameras and top camera are (u_i, v_i) and (u_t, v_t) respectively. A pair of image coordinates of side camera (u_i, v_i) and top camera (u_t, v_t) is combined into a 4 dimensional vector (u_i, v_i, u_t, v_t) which is used as the point if one. In the same way, a pair of the other side camera and top camera is combined into the input of another map. Since the valid workspaces of each map and camera are different, maps are used alternately. Besides the number of joint, no further information about the manipulator and cameras will be used in our visuo-motor system.

2.1 The self-organizing maps

As shown in Fig.2, each self-organizing map is consisted of neurons, which are distributed in the image spaces of the camera. Each neuron N_i has 4-parameters.

For ξ_i, refer to our previous study.

W_i : position of the neuron in two image spaces.

J_i : Jacobi matrix from the joint space to the image spaces.

θ_i : Joint angle of the manipulator at W_i.

ξ_i : The gradiant vector.

When a target u_t is given in the workspace, an appropriate map is chosen based on which cameras can see the target. In the chosen map, the neuron which w_i is the nearest to the projection of the target is chosen. The joint angles θ_{out}, which conduct end-effector to the target, are calculated obeying following linear equation. Although the transformation from the image spaces to the joint angle spaces is not a linear projection for a redundant manipulator, the domain of a neuron is small enough to use the linear projection as

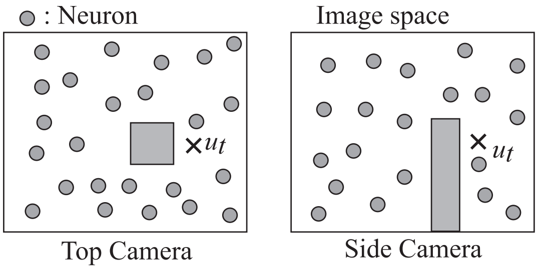

Fig. 2: Self-Organizing Map

an approximation of the non-linear projection. In the actual system, weighted sum of outputs from plural neurons around the target is used instead of the linear equation.

$$\theta_{out} = \frac{\sum g_n(\theta + J^{\dagger}(u_t - W))}{\sum g_n} \tag{1}$$

Where,$J^{\dagger}$ is a pseudoinverse matrix of J. g_n is the weight defined by the following equation.

$$g_n = \begin{cases} \exp(-n/\lambda) & \text{for } \exp(-n/\lambda) > \epsilon \\ 0 & \text{for } \exp(-n/\lambda) \leq \epsilon \end{cases} \tag{2}$$

Where, n is the order of the neuron determined according to the distance between the neuron and the target. It has a large value for the neuron that is near to the target, and has a small value for the neuron that is far from the target. The symbol λ and ϵ are value to define neuron numbers that can affect θ_{out}.

3 Process

3.1 Learning procedure of the Self-Organizing Map

In our system, plural SOMs are employed. If they are learned separately, their outputs are different even for the same target. This will result in that the manipulator moves inconsistently when it is driven from valid workspace of one map to another. In the learning algorithm, the problem has to be solved effectively. The problem also can be described as: the algorithm should guarantee that the manipulator moves smoothly and consistently in whole workspace no matter which SOM outputs joint angles for it.

The learning procedures illustrated in Fig.3. When a target position u_t is presented randomly in the workspace, each camera sees the target. If a camera can see the target, the SOM which is connected to the camera will learn. While more than two SOMs learn for a common visible target, they learn with influence each other. In this case, one of SOMs is chosen to determine the joint angles θ_{out} of the manipulator for the target. The manipulator is driven by the θ_{out}, and each camera obtains end-effector position v in each SOM respectively. Then, each SOM corrects its parameters by using u_t, θ_{out} and v. This learning procedure results in that in the end of learning the neurons of plural SOMs possess the similar value if $\theta_{out}, \xi$and different W, J for the target given in the common visible space.

Thus, the outputs from either SOMs will ensure the manipulator has the same pose. This means: while a target is given in common visible parts, the outputs from either SOMs do not result in a change of the manipulator pose. In addition, the assignment of similar joint angles to adjacent target point is, in fact, one of the main features of the learning algorithm of the SOM. By the construction of a map between inputs in the image space and the neural net, learning algorithm makes sure that adjacent target points always activate adjacent neuron in the network. The learning forces adjacent neurons to adapt their output towards similar values.

Therefore, at the end of the learning phase the output values will very smoothly from a neuron to another neuron. Both features bring about a continuous and smooth transformation from the input image spaces of target points to the out put space of joint angle sets. According to such a learning algorithm, SOMs guarantee smooth and consistent movements of the manipulator in whole workspace. For detials about evaluation function, learning algorithm and update of the parameter, refer to our previous paper [13].

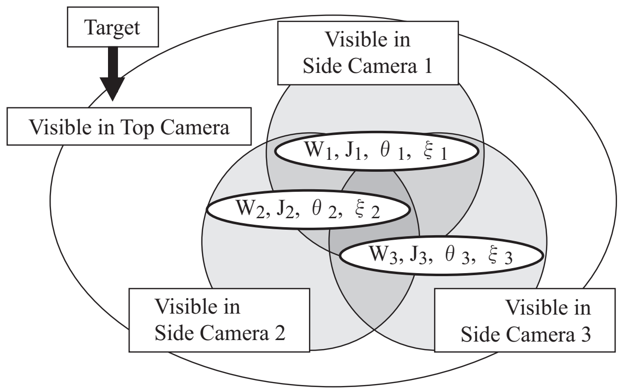

Fig. 3: Relation of Plural Self-Organizing Maps

3.2 Path planning

In our system, collision avoidance is based on the idea, While projected path does not interfere with projected obstacle, the path in 3 dimensional space can avoid the obstacle. Here collision avoidance is realized in the image spaces. It is not required to reconstruct a 3D model. Consequently collision avoidance in the image spaces by combining a SOM and a path planning system. The SOM determine joint angles of manipulator so that the end-effector of the manipulator reaches a target position given in the image spaces arbitrarily, and also ensures the manipulator taking obstacle-free poses. Here, the path planning system adapts Laplaces potential method to plan a collusion-free path for the end-effector from an initial to a goal position without plagued into local minima.

Since the manipulator can drive the end-effector to a target point given in the image spaces with obstacle-free poses under control of the SOMs, the path planning system only needs to plan a path for the end-effector of the manipulator. Accordingly, it is not necessary to pay attention to the collision between obstacles and links in the process of driving the end-effector along the planed path. It is different from most of existing algorithm, which are intended to work in configuration space. Our system always plans paths in 2D spaces, so the computational cost of planning does not increase exponentially for a high dimensional redundant manipulator. For procedure of the path planning and path planning by Laplaces potential method, refer to our previous study [12].

4 Simulation and experimental result

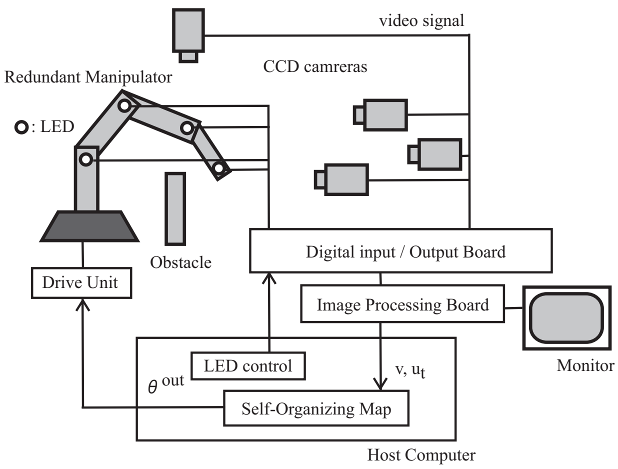

Fig. 4: Outline of experimental system

A photograph and the outline of our system are respectively shown in Fig.4 and Fig.5. The manipulator is Mitsubishi Manipulator RV-1A. It is driven by using the operating signal θ_{out} which is received through

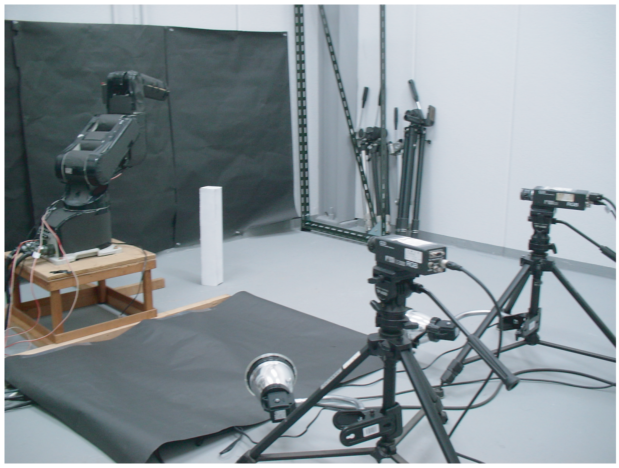

Fig. 5: Our experimental system

a Drive unit. The CCD cameras see the manipulator and the outputs from cameras are lead to a frame memory in the host computer. The size of the frame memory is 640[pixel] $\times$ 480[pixel]. The SOMs are composed in a host computer. LEDs, which are lighted by the host computer, are set on joints so that the system can find the position of the end-effector and links of the manipulator in camera images. Cameras are also used to get shapes of obstacles.

At first, by simulation, we will show that our system can output joint angles using 3 CCD cameras and 2 related SOMs so that the end-effector of the manipulator reaches to the target with obstacle-free poses, and that the system accomplish collision avoidance using proposed approach. Then, we will confirm reproductivity of the system to experiment in actual environment.

Secondly, we will expend the system to increase CCD cameras and SOMs and show that the system makes the manipulator take obstacle-free poses and accomplish collision avoidance.

Simulation results were shown in Fig.6 and Fig.7. In the simulation, a SOM consisted 240 neurons was used, and 15000 targets were given for learning. There was one obstacle. Fig.6 shows the target positions and the poses of the manipulator, when all cameras can see the target after learning. In the figure, we can see that the manipulator takes obstacle-free poses and the end-effector reaches targets correctly. The average error of the end-effector position was 1.57[pixel]. Fig.7 illustrates planed path from initial to goal point and manipulator poses driven by the SOMs without stopping at local minima. The average error along planed path was 3.06 [pixel].

An experimental result is shown in Fig.8 and Fig.9. To shorten the learning time and confirm robustness of the system, the SOMs of above simulation were

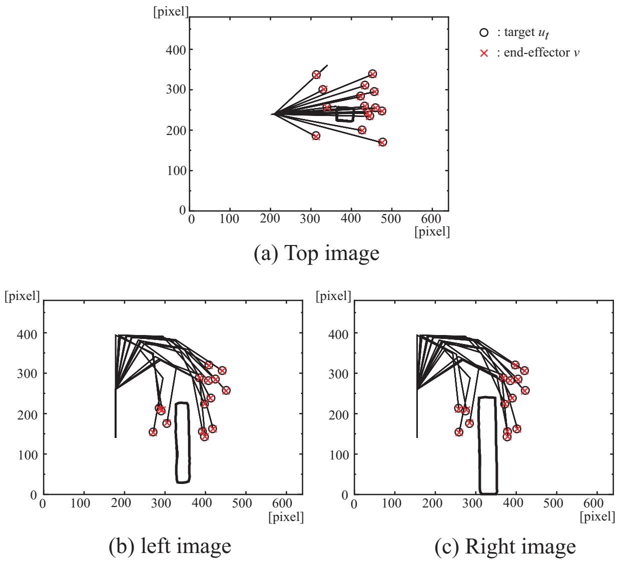

Fig. 6: Output for manipulator by simulation

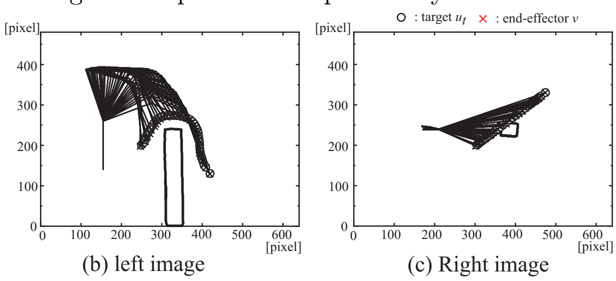

Fig. 7: Output of collusion avoidance by simulation

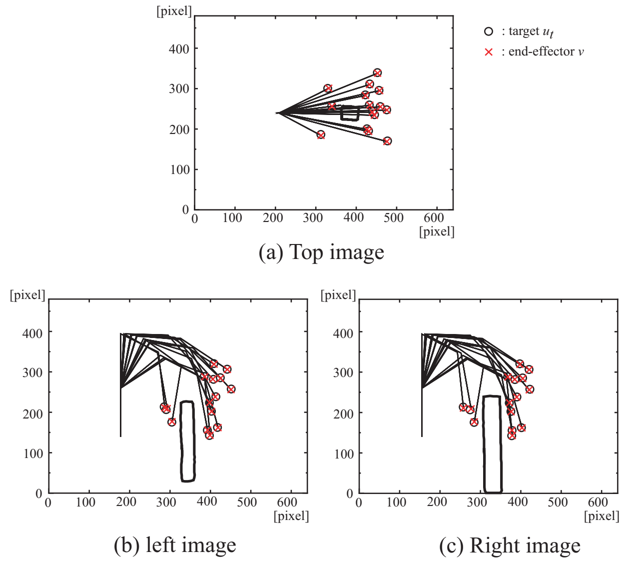

Fig. 8: Output for manipulator by experiment

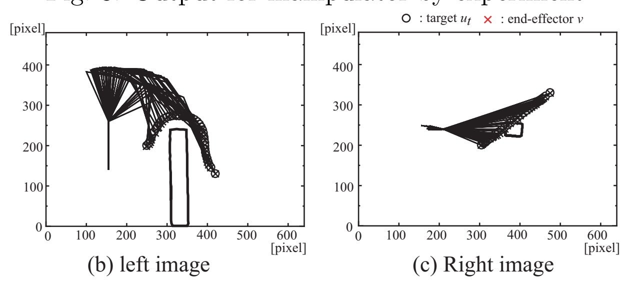

Fig. 9: Output of collusion avoidance by experiment

used as initial state by experiment. The number of re-learning time was 1500 and the time required process was about 16,500[sec]. Similarly with Fig.6, Fig.8 shows target positions and poses of the manipulator after re-learning. The average error of the end-effector position was 1.57[pixel], and it was about 2[mm] in the real environment. Also, Fig.9 illustrates the planed path and the manipulator poses. The average of error along planed path was 2.75[pixel].

By the simulation and experiments, we have verified that the system can make the manipulator take obstacle-free poses and practice collision avoidance.

Next, simulation results, which are leaded from the extended system, are shown in Fig.10 and Fig.11. In the simulation, a SOM consisted 240 neurons was used, and 30000 targets were given for learning. There was one obstacle. Fig.10 shows the target positions and the poses of manipulator, when all camera can see the target after learning. In the figure, we can see that manipulator takes obstacle-free poses and the end-effector reaches the targets correctly. The average error of the end-effector was 1.74[pixel]. Fig.11 illustrates planed path from an initial to a goal point and the manipulator poses driven by SOMs without stopping at local minima. The average of error along planed path was 2.56 [pixel].

5 Conclusion

In this paper, we validated the proposed method approach by real experiment and confirm the efficiency of the proposed method and robustness of the system. Then, we extended the system by increasing number of camera and SOM and showed the efficiency of the system. Advantages of this approach are (1) By employing multiple SOMs alternately, the system overcomes the occlusion problems in cluttered environment, (2) In our visuo-motor system, neither any priori knowledge about the manipulator nor the camera parameters are required.

In the present system, when a target point is put under obstacles and the top camera can not see it, SOMs can not learn joint angles because of dependence

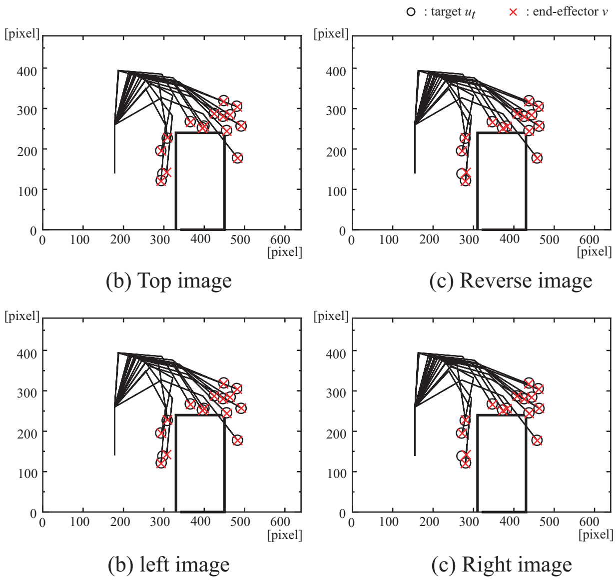

Fig. 10: Output for manipulator in extended system by simulation

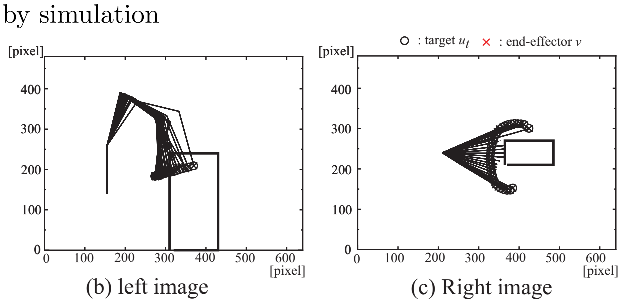

Fig. 11: Output of collusion avoidance in extended system by simulation

on the top camera. Therefore, we have to extend the system and make each camera independent.

A part of this study is financially supported by Electro-Mechanic Technology Advancing Foundation, Japan.

References

[1] S.M LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *Technical Report 98-11*,Computer Science Dept., Iowa State University, 1998

[2] J.F. Canny, “The complexity of robot mootion planning,” *MIT Press*, 1988

[3] R. Bohlin and L.E. Kavraki, “Path Planning using lazy PRM,” *In Proc.ICRA*, pp. 521-538, 2000.

[4] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Int. J. of Robotics Reseach*,5(1): 90-98, 1986

[5] K.Glass, R. Colbaugh, D. Lim and H. Seraji, “Real-time collision avoidance for redundant manipulators,” *IEEE Trans. On Robot. and Automat*, vol.11, no.3, pp-. 448-457, June, 1995

[6] H. Seraji and R. Colbaugh, “Improved configuration control for redundant robots,” *J. of Robot. Syst.*, vol.7, no.6, pp-. 897-928, 1990

[7] O.Khatib, S. Quinlan and D. Williams, “Robot planning and control,” *Robotics and Autonomous System*, 21: 249-261, 1997

[8] O.Brock and L.E. Kavraki, “Decomposition-nased motion planning: A framework for real-time motion planning in high-dimensional configuration spaces,” *In Proc. ICRA*, vol.2, pp. 1469-1474, 2001

[9] K.H. Low, W.K. Leow and M.H. Ang, Jr., “Integrated planning and cotrol of mobile robot with self-organizing neural network,” *In Proc. ICRA*, 2002

[10] S.X. Yang and M. Meng, “Neural network approaches to dynaic collision-free trajectory generation,” *IEEE Trans. Systems Man Cybernet*, B 31(3), pp. 302-318, 2001

[11] M. Zeller, R. Sharma and K. Schulten, “Motion planning of a pneumatic robot using a neural network,” *IEEE Control System Mag*, 17, pp. 89-98, 1997

[12] N. Okada, K. Minamoto and E, Kondo, “Collision avoidance for a visuo-motor system with a redundant manipulator using a sef-organizing map,” *ISATP*, pp. 104-109, 2001

[13] M. Han, N. Okada and E. Kondo, “Coorination of an Uncalibrated 3-D Visuo-Motor System Based on Multiple Self-Organizing Maps,” *JSME International Journal*, Series C, vol.49, No.1, pp. 230-239, 2006