Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A Novel Dynamic Priority-Based Action-Selection-Mechanism Integrating a Reinforcement Learning

Il Hong Suh
The Graduate School of
Information and Communications
Hanyang University,Seoul,Korea
Email: ihsuh@hanyang.ac.kr

Min Jo Kim
The Graduate School of
Information and Communications
Hanyang University,Seoul,Korea
Email: mjkim@incorl.hanyang.ac.kr

Sanghoon Lee and Byung Ju Yi
School of E.E.C.S.
Hanyang University,Seoul,Korea
Email: shlee@incorl.hanyang.ac.kr
bj@hanyang.ac.kr

***Abstract*— A novel action-selection-mechanism is proposed to deal with sequential behaviors, where associations between some of stimulus and behaviors will be learned by a shortest-path-finding-based reinforcement learning technique. To be specific, we define behavioral motivation as a primitive node for action selection, and then sequentially construct a network with behavioral motivations. The vertical path of the network represents a behavioral sequence. Here, such a tree for our proposed ASM can be newly generated and/or updated, whenever a new sequential behaviors is learned.**

To show the validity of our proposed ASM, some experimental results on a "pushing-box-into-a-goal(PBIG) task" of a mobile robot will be illustrated.

I. Introduction

An animat - either a stimulated animal or an animal-like robot - must select an action that is appropriate to the situation in which the animat lives and learns how to survive. Thus, an animat with sensors and actuators is usually equipped with a action selection mechanism (ASM) that relates its perception to its actions and make it possible to adapt its environment [1] [2].

One of the fundamental problems is to decide what to do next [3]. This problem is denoted as the action selection problem (ASP). In the ethological view, the ASP is the problem for an animal to design how to select its action so as to maximize its future expected genetic fitness [4]. But the ASP has proven to be a hard nut to crack due to (a) incomplete knowledge, (b) unpredictable environment and surrounding, (c) imperfect sensor and actuator, (d) limited resource [5].

The architecture of earlier systems, which were based on traditional AI planning methods, consisted of a sense-plan-act sequential cycle and the interaction between the sensing, planning, and action components. But traditional AI planning methods have some limitations, because They assume accurate knowledge of the world state provided by system sensors. This assumption is not valid due to a number of factors, such as changing world state, limited processing resources, and noisy, unreliable sensory information [5].

To overcome weakness of traditional AI approaches, a new reactive approach, called as "behavior-based AI", has emerged. Brooks [6] has suggested a new architecture being called "subsumption architecture", which are composed of competence modules with fixed priorities. This approach gives us an advantage, such as to fulfill a set of goals in a complex environment. To make an animat more life-like than subsumption approach, several researchers have proposed ethologically inspired model of action selection [5] [7]–[10]. Those models have showed good performances to imitate behavior of real life, since action selection in those models has been done based on competence modules with changing priorities. But most of those works generally involved 'fixed' pre-designed stimulus-response behavior systems and did not incorporate learning. Thus, they may not be appropriate in dynamic environments. Recently, several researchers has suggested ethologically inspired models of action selection that incorporate learning [11]–[13]. But much works remain to be done to cope with several shortcomings such as the lack of goal-handling ability and lack of learning sequential behaviors.

In this paper, we suggest a novel architecture that allows learning to be combined with action selection, based on ideas from ethology. Furthermore, we improve current ethology-based architectures to deal with sequential behaviors. Most of typical tree structures organize actions in a hierarchy that range from high-level "nodes" or activities via mid-level composite action to detailed primitive nodes. Thus, only the primitive actions are actually executable. Our proposed ASM, however, can select the most appropriate motivation in a given situation. And, our ASM can let a proper action be executed in every node within that motivation. As a result, Our ASM can choose correct sequential behaviors to satisfy a motivation and thus enables the system to learn necessary sequential behaviors.

II. Action Selection Mechanism

ASMs can be generally classified as arbitration or command fusion architectures [15].

Arbitration mechanisms select one behavior, from a group of competence modules. Arbitration mechanisms for action selection can be divided into : fixed priority-based, winner-take-all, state-based. In fixed priority-based mechanisms, an

action is selected based on a priori assigned priorities [15].

The subsumption architecture proposed by Brooks [6] is typical fixed priority-based mechanism. This architecture consists of a series of behaviors, which constitute a network of hardwire finite state machines. Action selection consists of higher-level behavior overriding the output of lower-level behavior. Thus, each competence module of a level can be considered as having a priority, and high priority module suppresses low priority module. The control system is hardwired directly in the structure of the behaviors and their interconnections, and can thus not be altered without redesigning the system. This type of architecture can be called as fixed priority-based arbitration architecture.

The other type of arbitration architecture is winner-take-all architecture, which is more flexible than subsumption architecture. Maes [9] and Blumberg [11] suggested this type of architecture. In this mechanism, action selection results from the interaction of a set of distributed behaviors that compete until one behavior wins others. Each competence module is considered to have priority varying under its own external and internal influences. Because these mechanisms are more flexible than fixed priority-based architecture, learning process can be easily incorporated.

Blumberg [11] suggested an architecture that allows learning to be combined with action selection, based on ideas from ethology. But, their work mainly focused on "do the right thing in a given situation". Therefore their structure only selects a single behavior to satisfy its need, and learns simple S-R associations. Note that behaviors to achieve a mission are consisted of a series of behaviors. Selecting a single behavior in a given situation is not enough to accomplish a mission.

Contrary to Bulmberg's model, our proposed ASM can decide both "what to do next?" and "how that work can be achieved?". For this, a motivation that denotes a mission or goal competes with other motivations on the basis of internal needs and external stimuli. Next, a specific behavior to satisfy winner-motivation will be selected.

Our proposed ASM is a hierarchical organization of primitive modules named as Behavioral Motivation (BM) having their own stimuli(sensors) and behavior. To be more specific, we divide BMs into two types. Fist type is the Static BM(SBM) that denotes a motivation or mission. The connections among SBMs are fixed and cannot be changed until it is redesigned. Second type of BM is Dynamic BM(DBM), which can generate and learn sequential behaviors to satisfy the motivation.

Fig. 1 illustrates block-diagram of our ASM. In Fig. 1, Perception Filter(PF) system is a group of PF that filters external world information, and Internal State(IS) system is a group of internal influences such as drives. Fixed Action Pattern(FAP) is a series of actions to be also termed as behavior. FAP system is a group of FAPs. Thus, BM has a link among PF, IS, and FAP. Finally, Learning system stores the information of past stimulus-action pairs, and it computes values of taking the action in the situation. Learning system enables the animat to learn new sequential behaviors, and to add these sequential behaviors to the BM system.

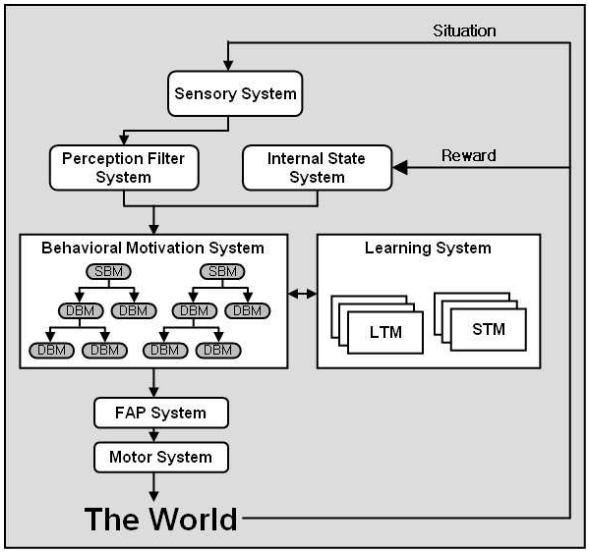

Fig. 1. Overall ASM architecture

A. Static Behavioral Motivation (SBM)

An SBM implies a mission or a motivation, and values of SBMs will be used to determine what a motivation would be activated for given external stimuli and internal needs. For this, value of each SBM is computed by combining the value of IV with that of PF, and then by inhibiting other values of SBMs. In addition, an SBM receives a feedback effect from the DBM group under this SBM. The value of an SBM is calculated by using the equation given by

$$V_{SBM_{i(t+1)}} = \sum V_{IV_{jt}} + \sum V_{PF_{kt}} - \sum_{allSBM} (V_{SBM_{lt}} I_{il}) + effect^{i}_{DBM}, \quad (1)$$

i : index of where the i th SBM
j : index of related IS
k : index of related PF
I_{il}: inhibitory Gain that SBM_i applies against SBM_l
l : index of same level SBM(inhibition).
$effect^{i}_{DBM} = w_i V_{DBM_m}$
m: index of maximum valued DBM under i th SBM
w_i: weight of feedback value from DBM under the i th SBM

The term $effect^{i}_{DBM}$ means the strength indicating how the goal can be easily achieved for a given current state of the environment. Thus, the value of an SBM may be high not only when needs of the SBM become more important than those of other SBMs, but also when its goal is believed to be easily achieved for the given current state of the environment. Each SBM has a group of DBMs implying sequential behaviors to satisfy the SBM(or motivation). SBMs are organized into a pre-designed flat-network as in Fig. 2.

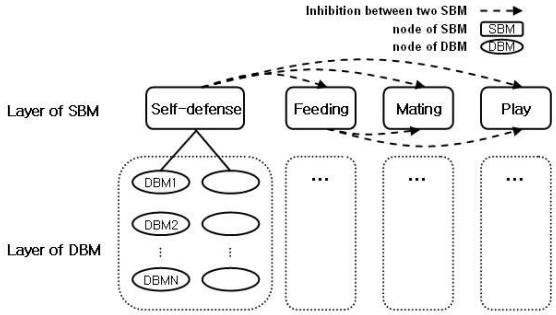

Fig. 2. An examplar configuration of SBM and DBM

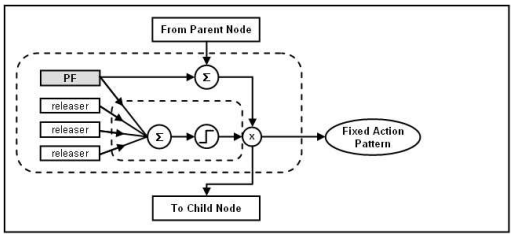

Fig. 3. A primitive node of a DBM

B. Dynamic Behavioral Motivation (DBM)

To accomplish a goal(or mission), an animat must generate a series of behaviors and select the most appreciate one. For this, DBMs are organized into flexible hierarchical network that can be changed by learning process. A DBM has its own activation-value that depends on the values from PF, parent node, and releasers. A DBM outputs its value to child node, while relevant stimulus is incoming. The schematic of a DBM is illustrated in Fig. 3. The activation-value is accumulated through the path, while relevant stimulus presents. Releasers play a role of blocking the flow of activation-value. The value of a DBM is given as

$$V_{DBM_i} = (V_{DBM_i-1} + V_{PF_i})STEP\left(\sum_{k=1}^{m} V_{Releaser_k}\right), \quad (2)$$

i: index of this node
j: index of related PF
k: index of related releaser

$$STEP(x) = \begin{cases} 1, & \text{for } x > 0 \\ 0, & \text{for } x = 0. \end{cases}$$

The appropriate DBM will be selected by choosing maximum-valued DBM in a DBM group. Following is the equation to select a DBM;

$$selectedDBM = \underset{i \in all\mathrm{DBM}under\mathrm{SBM}}{arg\ \max} (DBM_i). \quad (3)$$

The path from the top level DBM to the bottom level DBM consists of sequential behaviors. By performing these sequential behaviors, the motivation (or SBM) can be satisfied.

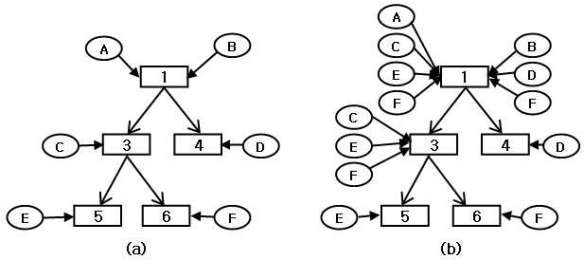

Fig. 4. An example of sensory bottleneck

C. Input variables for SBM and DBM

Perception Filter(PF): A Perception Filter (PF) identifies significant stimuli or events from input sensors, and output a value, which represents its strength and relevance. In other words, each PF outputs a continuous value which typically depends on existence of specific stimuli, and some quantitative measure such as distance. SBM and DBM can be made more or less sensitive to the presence of the relevant stimulus.

Releaser: In our proposed ASM, the DBM will convey its value to the child node, while relevant stimulus is incoming thru corresponding PF. But, the flow of activation-value will be blocked, if relevant stimulus disappears. That is, relevant stimulus plays a role of releasing activation-value. This PF, which releases and/or blocks the value-flow, is called as 'releaser'. By releasers, our proposed ASM can deal with sequential behaviors. Fig. 4 shows sensory bottleneck [5]. In order to activate the node 5 stimulus E is required. But the node 1 and the node 3 does not pass stimulus E. Therefore, the node 5 may not be activated. To avoid such a blocking of stimuli, every nodes must pass the stimulus to their lower nodes, which is valid in Our ASM by use of releasers.

Internal State: An Internal State (IS) is used for modeling drives such as hunger or thirst. The FAP, which reduces a certain IS or satisfies drives, is called as consummatory action and other FAPs are called as appetitive actions. The value of IS is changed after an FAP is executed. The relation of FAP and IS is defined by the gain. Based on Hull's theory [16], the reduction of drive can lead to a learning process. Therefore, the reduction in the value of IS by executing a consummatory action is used as a reinforcement signal for the learning. The value update equation of IS is given by

$$IS_i = IS_i + gain_{ij}, \quad (4)$$

i: index of IS
j: index of selected FAP .

D. Action Selection Process

Our ASM selects the most appropriate behavior(or FAP) based on its internal need and external environment on every cycle. To do this, the BM system selects relevant SBM and DBM. The SBM means what a mission must be achieved, and the DBM means what an action must be selected to satisfy a given motivation. The value of a SBM(or mission) depends not only on its internal needs, but also on how easily the goal will

be achieved. Thus, the value of a SBM reflects their internal need and feedback effect from its DBM. The selecting process for SBM and DBM is summarized as follows;

- A maximum-valued DBM is selected in each DBM group.
- Value of each SBM is computed by using (1), and then is compared with those of other SBMs to select the maximum-valued SBM.

After a maximum-valued SBM is chosen, one of the following two processes will be activated to select an action; 'exploit' and 'explore'. An exploit-process is performed when a BM system has enough knowledge to satisfy its motivation. The exploit-process is performed by executing the most appropriate FAP(or behavior) for a given situation.

Otherwise, an explore-process is performed (i) when a BM system has no knowledge to satisfy its motivation, or (ii) when a BM system has a little knowledge to satisfy its motivation. Especially, situations with no prior knowledge are divided as follows;

- When the selected SBM has an empty DBM group
- When the selected SBM does not include a DBM that could be matched with the current situation
- When a DBM is selected several times without reaching a goal

Like an exploit-process, an explore-process should decide an action. Specifically, a PF is randomly selected among PFs that have non-zero values. An FAP is also randomly selected among all FAPs. Then, the selected FAP is executed and the selected PF and FAP will be stored in STM. After several cycles, if an animat satisfies the given motivation, past relations between PF and FAP that have been stored in STM will be transferred to LTM as will be described in section III-A.

Now, when a little prior knowledge is available for an SBM, such a knowledge, which will be a group of relations of PFs having non-zero values and all FAPs, will be involved to select an action. That is, one relation of PF and FAP among a group of known relations will be selected and the corresponding FAP will be chosen based on the probability given as the reliability value of LTM. Fig. 5 illustrates overall process of our action selection. Here, it is remarked that to learn different strategies for different motivations, each SBM has its own LTM and STM.

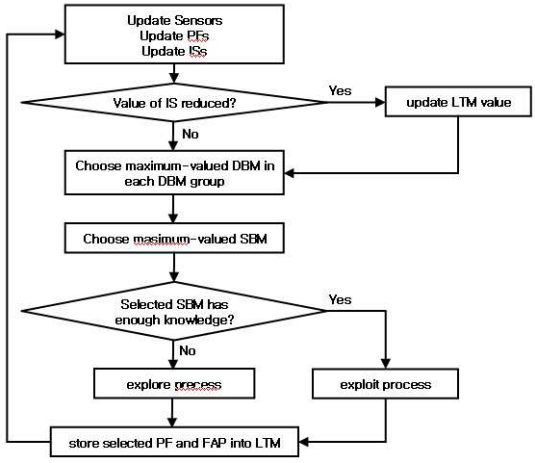

Fig. 5. The process of Action Selection

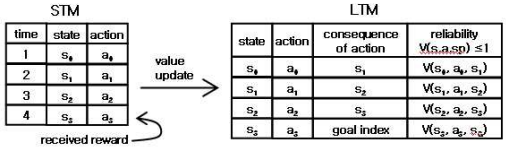

Fig. 6. STM and LTM

III. Integration of Learning into ASM

A. Learning Process [17] [18]

In this paper, a RL algorithm involving shortest-path-finding algorithm is employed, where internal reasoning process is incorporated to reduce probability of selecting wrong behaviors. Specifically, first, RL is applied until a successful episode is completed. Then, all S-R behaviors of the episode, which have been recorded in a short-term memory (STM), are transferred to long-term memory (LTM), in which a shortest path-finding algorithm is applied to find out the shortest path from initial S-R behavior to the goal stimulus. S-R behaviors on the shortest path are the highly credited by increasing their reliability values, and S-R behaviors which are not on the shortest path are punished by reducing their reliability values. Since an S-R behavior will be selected based on behaviors and their reliability value of LTM, it is expected to reduce the probability of choosing unnecessary behaviors, and thus to increase the learning speed.

Internal memory consists of short-term memory (STM) and long-term memory (LTM). In the STM, stimulus-response (S-R) pairs are recorded along the time. Fig. 6 shows a process in which data in STM are moved to LTM. Once all data in STM is moved to LTM, then STM is cleared to store data of new exploration. All S-R pairs in STM are assumed to be appetitive behaviors to get rewards. With references to the time at which a reward is received, S-R behaviors near the reference time should be more credited than S-R behaviors far from the reference time. This can be reflected as a reliability value, V_{ij}, given by

$$V(s_i, a_i, s_{i+1}) \leftarrow V(s_i, a_i, s_{i+1}) + \frac{\eta(1 - V(s_i, a_i, s_{i+1}))}{i^{\lambda}}, \tag{5}$$

where s_i is the index of the i^{th} stimulus, a is the index of response behavior, η is a learning rate, λ is decay rate, and i^{λ} is weightings of distance from reference time. It is remarked that (5) is similar to equation for Monte-Carlo method. Fig. 7 shows pseudo-code of STM and LTM operations for exploration of reinforcement learning.

repeat(for each episode)
  initialize STM
  repeat(for each step of episode)
    choose action and state **a**, **s** with respect to $V(s,a,sp)$ (Boltzman exploration)
    store **a**, **s** to STM
    take action **a**
LOOP i = 1 to STMsize
  IF i = STMsize
    $V(s_i, a_i, \text{GoalIndex}) = 1$
  ELSE
    $V(s_i, a_i, s_{i+1}) \leftarrow V(s_i, a_i, s_{i+1}) + \eta \frac{1 - V(s_i, a_{i+1}, s_{i+1})}{i^{\lambda}}$

$V(s,a,sp)$: LTM value
sp : consequence of action
η : learning rte $0 < \eta < 1$
λ : decay rate $0 < \lambda < 1$

Fig. 7. Pseudo code of suggested learning algorithm

B. Integration of Learning

After the animat performs some sequential behaviors and receives a reward, stimulus-action pairs that consist of sequential behaviors will be stored in the LTM. Thus, the LTM may include several paths to accomplish task. LTM entries with values, which exceeding a certain threshold will be added to BM system. Fig. 8 shows an example of interaction between the Learning system and the BM system. Note that Fig. 8(a) shows the LTM state after some trials were performed, and the value of 'S_G–B_G–goal' exceeds a threshold. Because the LTM entry 'S_G–B_G–goal' has not been included in corresponding hierarchical structure, this entry will be added to the branch of the SBM.

After more trials are performed, the LTM may be changed as in Fig. 8(b). In Fig. 8(b), there are two LTM entries with values to exceed the threshold. Because the entry ('S_G–B_G–goal') has been already included in the hierarchical structure, another entry ('S_3–B_3–S_G') will be added. To reach the goal, the action B3 in 'S_3–B_3–S_G' must precedes the action in 'S_G–B_G–goal'. Thus, the position of that entry will be the parent node 'S_G–B_G–goal'. After many trials, a new appetitive behavior to reach goal can be added in the structure as shown in Fig. 8(c).

IV. Experiments

A. AmigoBot System

AmigoBot of ActivMedia Robotics Company [23] is employed for our experiments as shown in Fig. 9. The robot has 8 sonar sensors and one CCD camera. Pentium PC(450MHz) is used to control the robot, where 900MHz RF modem is utilized to communicate with robot, and 2.2GHz A/V receiver is used to get the video data of CCD camera.

B. Experimental Objectives

To show the validity of our proposed ASM integrating a reinforcement learning, experimental set-up for a pushing-box-into-a-goal(PBIG) task of a mobile robot(AmigoBot of Active Media Robotics company [23]) is organized as shown in Fig. 10.

In this experiment, a mobile robot is shown to learn sequential behaviors to push a box into a goal located outside the fence by our reinforcement learning approach. Whenever

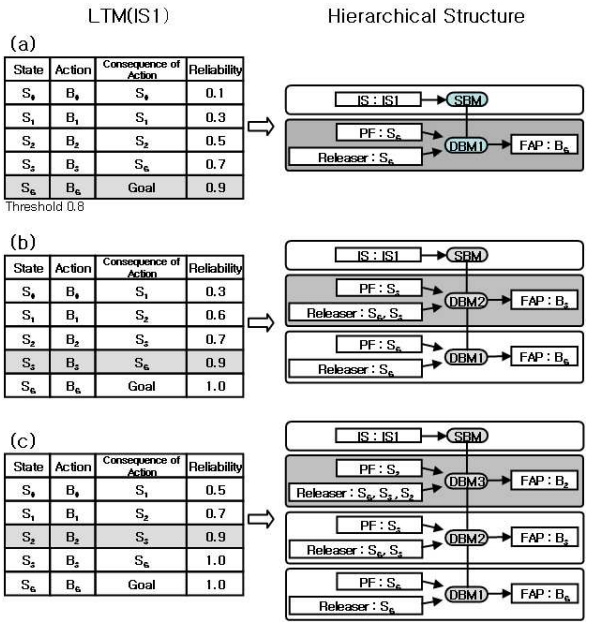

Fig. 8. An example of interaction between LTM and hierarchical structure

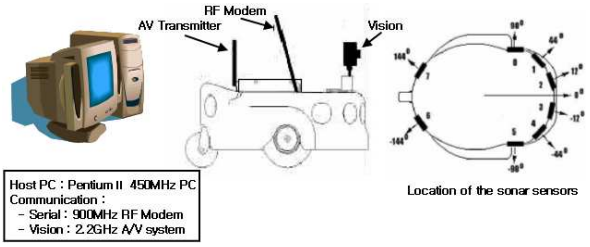

Fig. 9. AmigoBot System

the robot completes the PBIG task successfully, the robot receives a reward. And then, those learned sequential behaviors responding to the input stimuli are shown to be automatically on-line recorded as sequential DBMs in our proposed ASM.

C. Fixed-Action-Patterns and State-Organization

FAP and state organization are summarized as in Table I and Table II, respectively. It is known from Table I that 22 Fixed-Action-Patterns(or termed as behaviors) are employed. Among them, there are 6 moving behaviors with different durations, and 12 turning behaviors, and 4 PBIG task-related behaviors. Especially, SearchTarget1 is the behavior to change heading

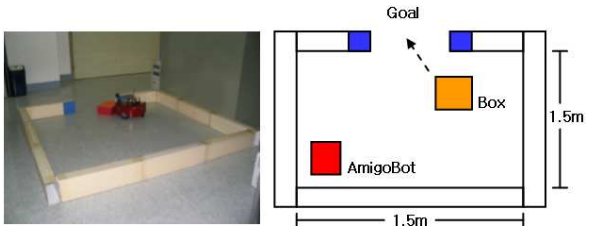

Fig. 10. Experimental Set-up for a PBIG task of a mobile robot

TABLE I

FIXED ACTION PATTERNS(BEHAVIORS) USED IN EXPERIMENTS

NAME	VELOCITY (mm/sec)	HEADING (degree)	DURATION (msec)
MoveF_Fast	200	0	2000
MoveF	100	0	2100
MoveF_Slow	50	0	2000
MoveB_Fast	-200	0	2000
MoveB	-100	0	2100
MoveB_Slow	-50	0	2000
TurnR_15	0	-15	0
TurnR_30	0	-30	0
TurnR_45	0	-45	0
TurnR_60	0	-60	0
TurnR_75	0	-75	0
TurnR_90	0	-90	0
TurnL_15	0	15	0
TurnL_30	0	30	0
TurnL_45	0	45	0
TurnL_60	0	60	0
TurnL_75	0	75	0
TurnL_90	0	90	0
Push	100	0	2000
SearchTarget1	0	-30	0
Approach	100	0	2100
SearchTarget2	0	-30	0

TABLE II

STATE ORGANIZATION FOR EXPERIMENTS

Variables	Description
Target Position	Box Position (1 : Left, 2 : Middle, 3 : Right, 4 : Unknown)
Target Shape	Box Shape (1 : —, 2 : /, 3 : \, 4 : Unknown)
Target Distance	Box Distance (1 : Far, 2 : Near, 3 : Close, 4 : Unknown)
Target2 Position	Exit Position 0 ~ 23, -1 : Unknown

angle by 30 degree in counter-clockwise direction without running. And, SearchTarget2 is the behavior to change heading angle by 30° in the counter-clockwise direction without running. Table II explains how we organize the environmental state. To be specific, there are 4 box locations including unknown location with respect to the mobile robot, 4 box shapes including unknown shape from the camera view of the mobile robot, 4 types of distance to the box from the mobile robot, and 24 directions to the target2 position from the location of pathway to the goal outside the fence

D. State Recognition by a color CCD camera and a multilayer neural network

In this experiment, a color CCD camera image of 320×240 pixels with 24bits color data is employed to discriminate

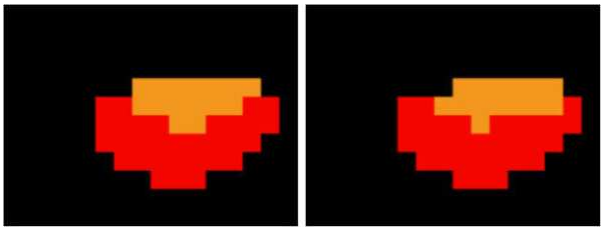

Fig. 11. Two different 16×12(192) perceptual input images for the almost same environment

the environmental state observed by the mobile robot. It is here remarked that we first did not use high-level computer vision technique to avoid high computational burden. Instead, we employ a low level vision data processing as follows: for effective clustering of similar states and for real-time computation, image data of 320×240 is smoothed by using 192 windows where the size of each window is given as 20×20 pixels. And, 192 representative color values from 192 windows are extracted by low level image processing techniques such as thresholding, segmentation, and a representative color value extraction. They are utilized as an input perceptual vector, where the representative color value will be one of orange(1), red(2), blue(3), and background(-1). It is here remarked that we could have quite different input perceptual vectors for the almost same environmental state due to time-varying lighting condition, noise, and slight change of relative distance and orientation as shown in Fig. 11. This might cause us to get the perceptual aliasing problem. It is pointed out in [19] that the perceptual aliasing problem could make the mobile robot haunted since boxes often looked like walls, or relative location and orientations of the box could not be well-differentiated in a low-level CCD camera image. To cope with such a perceptual aliasing problem, a multilayer perceptron neural network is employed, where numbers of input nodes, hidden nodes, and output nodes are chosen as 192, 20, and 10, respectively, as shown in Fig. 12. Input perceptual vector of 192 dimension is used as input for the neural network. And followings are employed as output nodes; 3 relative-box locations with respect to the mobile robot, 3 box shapes observed by the mobile robot, 3 distance types to the box, and yes or no for finding the pathway to outside. Thus, number of clusters for the neural network to classify will be 54(3×3×3×2) 923 images takes at 54 states were used to train the network. Fig. 13 shows a successful result of recognizing an environmental state by the neural network after training.

E. Experimental Results

A reward signal is given only when the mobile robot completely get a success to push the box to the goal location outside the fence through the pathway without collision with the fence. First, to see that our robot learn how to do the mission by using a simple monolithic(single) DBM tree, 200 trials were performed, but unfortunately this approach failed to get even a single success. Here, each trial was organized to last all steps which was generated from the beginning to

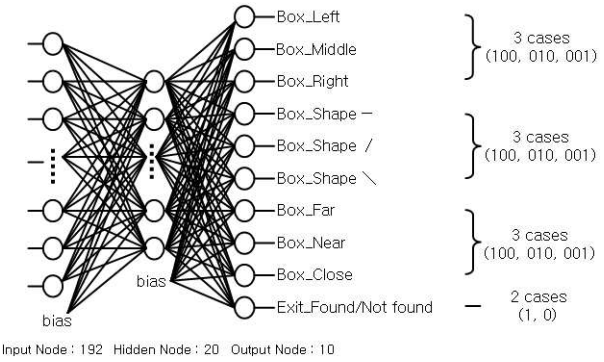

Fig. 12. Proposed multilayer neural network

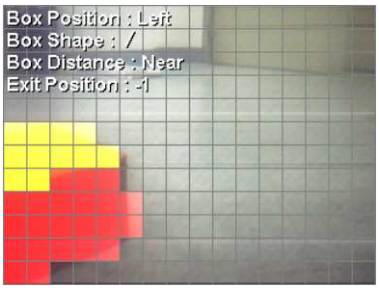

Fig. 13. Successful recognition of a environmental state(Box_Left, Box_shape"/", Box_Near, and Exit_Not found) by the proposed multilayer neural network

the instant of collision with the fence. In an averaged sense, the robot takes 100 steps in a learning trial. At the beginning of each trial, the world is restored to its initial state as shown in Fig. 10. This includes resetting the boxes and the robot to their initial locations. It is here noted that a box-push-into-the-goal task actually involves several distinct subtasks(or submissions). And, many of learning problems are expected to be solved, if a robot is made to learn each subtask(or submission) separately. Similar approaches were taken in behavior-robotics [20] [21] [22]. But in their methods, fixed-priority mechanisms such as subsumption were employed, and thus, priority should be given before learning. And Q-maps were learned in their approaches instead of directly learning a sequence of associations between states and behaviors. Therefore, our push-box-to-goal task is made to involve following three subtask; (A) the robot needs to find the potential box(searchTarget1) and approach to the box(approach) Also, the robot needs to find the pathway to the goal(searchTarget2). These first 3 BM's are included in the BM tree as innate knowledge of the robot. (B) The robot needs to be able to push a box through the pathway. This might be not easy because a box tends to slide and rotate unpredictably. (C) The robot needs to be able to approach to the easy-to-push location at which the robot may successfully push the box through the pathway as of subtask(B). Now, we will first take a look at how will the robot learns how to do the PBIG task(mission) with innate BM's for the subtask(A). Among 200 trials, only one trial is found to be a success, even though the robot effectively found the box and approach the box. This shows that subtask(B) and subtask(C) are hard to learn in a monolithic way due to an explosive number of explorations.

Fig. 14 shows experimental organization to learn how to do the subtask(B). It is noted that to push the box outside the fence through the pathway, at least 3 times of push-behavior should be generated. After 70 trials, we get a success, and then get another success after 10 trials. After all, an effective BM is learned as shown in Fig. 15. As depicted in Fig. 15, in the sequel, PF and corresponding behavior of DBM are only displayed for the sake of brevity.

Fig. 16 shows experimental organization to learn how to do the subtask(C). In this case, a reward is given if the robot moves from location 1) to location 2) without collision with the box in such a way that box shape is seen as"–". And, experimental world is restored to its initial state, whenever the robot pushes the box. Because location 1) is very near the box, the robot often pushes the box or collides with the box, while exploring correct behaviors. And, first success comes out after about 120 trials. Another success comes out after about a few trials. Fig. 17 and Fig. 18 shows BM tree after learning subtask(C). 3 parallel BM trees has been tested to know how well our global PBIG mission as shown in Fig. 10 can be completed with such learned BM trees. From Table III, we can observe that more than 80% of trials were successful. It is here remarked that the performance can be made better, since our dynamic BM approach can replace better rules with old ones after re-learning.

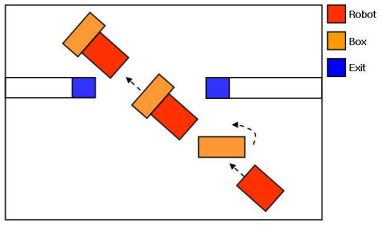

Fig. 14. Subtask(B): Push-Box-Thru-The-Pathway

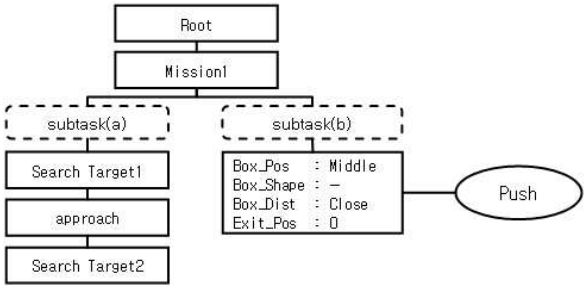

Fig. 15. A DBM learned for push-box

V. Conclusion

Selecting and learning appropriate actions to survive in its environment are the most important abilities in a robot. For this, we proposed a hierarchical organization of competence

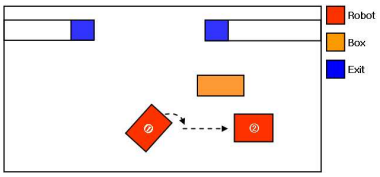

Fig. 16. Subtask(C): Approach to Easy-To-Push location

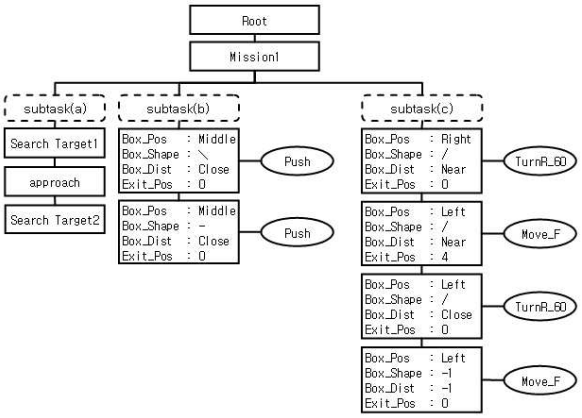

Fig. 17. BM tree for pushing-box-into-a-Goal task after learning

modules called as SBM and DBM. The SBM was used to select the most appropriate motivation in a given situation. And, the DBM was used to select a behavior that could satisfy its motivation. By use of releasers to pass or block the activation-value of DBM, a DBM tree can generate sequential behaviors. Thus, our proposed ASM can not only select the most appropriate behavior in a given situation, but also deal with sequential behaviors. Furthermore, our proposed flexible tree network enables the robot to add newly learned behaviors, or to delete unuseful old behaviors for better performance.

Experiments for a pushing-box-into-a-goal task of mobile robot was performed, where necessary BM trees were satisfactorily learned in the sense that performance of such BM trees was measured as 80% success among 100 PBIG task trials.

ACKNOWLEDGMENT

This work has been supported in part by Next-Generation New Technology Development Program entitled as Control/Recognition Technology for Personal Robots and 21C Frontier Program for intelligent robotics.

REFERENCES

[1] A.Guillot and J.A.Meyer,"Computer simulations of adaptive behavior in animats,"Proceedings Computer Animation'94.,IEEE Computer Society,1994.
[2] S.W.Wilson,"The Animat Path to AI,"In From Animals to Animats, First International Conference on Simulation of Adaptive Behaviour,MIT Press,Cambridge,1991.
[3] P.Maes,"Modeling Adaptive Autonomous Agents,"Artificial Life Journal,Vol.1,No.1&2,pp.135-162,MIT Press,1994.
[4] P.Pirjanian,"An Overview of System Architectures for Action Selection in Mobile Robotics,"Tech-report,Laboratory of Image Analysis,Aalborg University,1997.

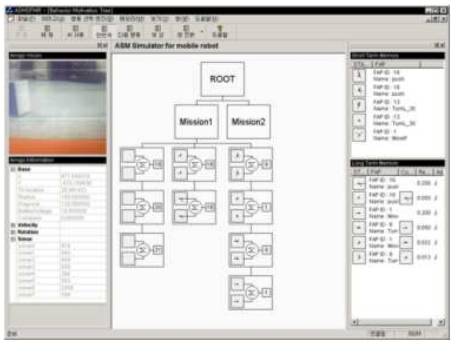

Fig. 18. Monitor display of STM, LTM, and BM tree after learning PBIG task

TABLE III
PERFORMANCE OF PUSHING-BOX-INTO-A-GOAL TASK AFTER LEARNING

Mission	Trial	Success	State Misjudgment	Robot Error
Box Pushing	100	81	15	4

[5] T.Tyrrell,"Computational Mechanisms for Action Selection,"Ph.D. Thesis,Centre for Congitive Science,University of Edinburg,1993.
[6] R.Brooks,"A Robust Layered Control System For a Mobile Robot,"In IEEE Journal of Robotics and Automation,pp.14-23,April,1986.
[7] J.K.Rosenblatt and D.W.Payton,"A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control,"Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, Vol.II,pp.317-323,1989.
[8] E.Spier and D.McFarland,"A Finer-Grained Motivational Model of Behaviour Sequencing,"In From Animals to Animats 4: Proceedings of SAB96,1996.
[9] P.Maes,"A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature,"In From Animals to Animats,First International Conference on Simulation of Adaptive Behaviour,MIT Press,Cambridge,1991.
[10] B.M.Blumberg,"No Bad Dogs: Ethological Lessons for Learing in Hamsterdam and Interactive Createures,"MIT,1997.
[11] B.M.Blumberg,"Old Tricks, New Dogs: Ethology and Interactive Createures,"MIT,1997.
[12] C.M.Witkowski,"Schemes for Learning and Behaviour: A New Expectancy Model,"Ph.D.Thesis,University of London,1997.
[13] M.Humphrys,"Action Selection methods using Reinforcement Learning,"Ph.D.Thesis,University of Cambridge,England,1996.
[14] D.C.Mackenzie,R.C.Arkin,and J.M.Cameron,"Specification and Execution of Multiagent Missions,"Autonomous Robots,4(1),1997.
[15] P.Pirjanian,"Behavior Coordination Mechanisms - State-of-the-art,"Tech-report IRIS-99-375,Institute for Robotics and Intelligent Systems,School of Engineering,University of Southern California,Oct.,1999.
[16] C.L.Hull,"Principle of behavior,"new York: Appleton-Century-Crofits,1943.
[17] W.Y.Kwon,S.Lee,and I.H.Suh,"A Reinforcement Learning Approach Involving a Shortest Path Finding Algorithm,"IEEE Intl.Conf.on Intelligent Robots and Systems,pp.436-411,2003.
[18] I.H.Suh,S.Lee,B.O.Kim,B.J.Yi,and S.R.Oh,"Design and Implementation of a Behavior-Based Control and Learning Architecture for Mobile Robots,"IEEE Intl.Conf.on Robotics&Automation,pp.4142-4147,2003.
[19] S.D.Whitehead and D.H.Ballard,"Active Perception and Reinforcement Learning,"Proc. of the 17th Intl. Conf. On Machine Learning,June 1990.
[20] R.A.Brooks,"A robot that walks: Emergent behvior form a carefully evolved network,"Neural Computation,1(2):253-262,1989.
[21] J.H.Connel,"A Behavior-Based Arm Controller,"IEEE Tran. On Robotics and Automation,5(6):781-791,1989.
[22] J.H.Connel and S.Mahadevan"ROBOT LEARNING,"Kluwer Academic Publishers,June 1993.
[23] "AmigoBot User's Guide,"ActiveMedia Robotics Company,2000