Key words – robot, behavior, intelligent, Bayesian network, probabilistic approach

So-Youn Park and Ju-Jang Lee

PROBABILISTIC DECISION MAKING OF ROBOT BEHAVIOR BASED ON BAYESIAN NETWORK

This paper presents a technique for an intelligent software robot (sobot) Rity to behave in uncertain environment in an appropriate manner. The intelligence of a robot is necessary to infer an appropriate behavior when various sensor data (stimuli) exist simultaneously and a specific behavior can be decided by a behavior controller. There could be various methods to build a behavior controller. Here we make behavior scenarios to set priorities among behaviors and use sobot Rity and Bayesian network to model an intelligent behavior controller and to make a certainty factor of behavior candidates by probability computation. Simulation results show that Rity behaves well following behavior scenarios with the proposed architecture.

1. INTRODUCTION

For decades, robots have been widely used in our daily life. Nowadays, with intelligence, robots even can interact with the human being. Also they can act in virtual world as well as in real environment. There are many ways to develop an intelligent robot such as using neural networks or fuzzy inference systems.

Meanwhile, Bayesian network is based on probabilities and can handle uncertain circumstances. Imamura introduced behavior learning method based on Bayesian Networks and experience of interaction between human and robots[1], Go made intelligent service models using Bayesian networks[2], Min developed the intelligent robot avoiding obstacles using behavior network and Bayesian inference in two levels[3].

In this paper, based on behavior scenarios designed in advance, we are going to develop a behavior controller of Rity using Bayesian network. The remaining of our paper is organized as follows. In session 2 we introduce Rity[6][7] and mention behavior scenarios. A basis of Bayesian network for building behavior controller and the designed behavior controller using Bayesian network is explained in session 3. We show the experimental results and discuss them in session 4 and finally conclude the work and suggest the future work.

2. OVERVIEW

Rity is a software robot which consist of 47 sensors having 4 states and 73 actions in $600\times600(\text{pixel}^2)$ virtual environment. It contains 3 objects and 1 user interface.

2.1. STRUCTURE

An architecture of Rity consisting of some sub-systems is shown in Fig. 1. Different from original architecture[7], we omit learning system for convenience and change internal status system somewhat. In perception system, there are various sensors that we can get direct sensor data and infered information from direct data. Our greatest concern, behavior controller, is included in the behavior system.

In this paper, we divide internal status system into two sub systems: motivation and emotion. Motivation system consists of 3 features: Boredom, Hunger, and Fatigue and emotion system consists of 4 features:Happy, Angry, Sad, Fear. Magnitudes of features increase depending on the sensor data(stimuli) and decrease depending on time. Fig. 2. shows Rity in virtual environment with visual sensor enable to be shown.

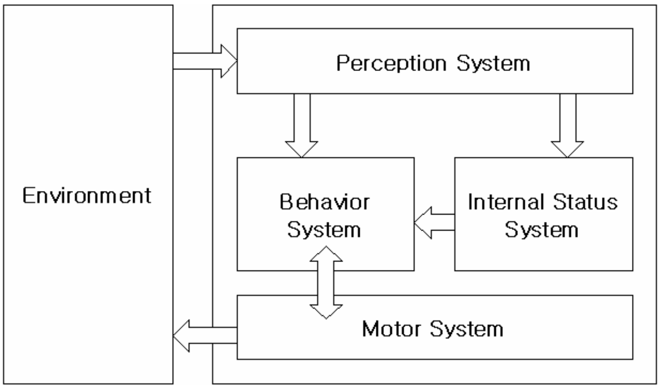


Fig. 1 An architecture of Rity

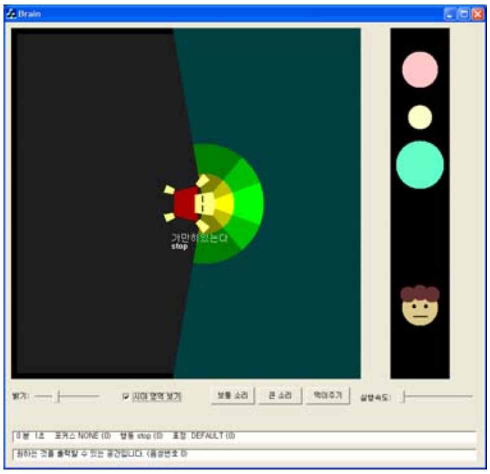


Fig. 2 Rity in virtual environment

2.2. BEHAVIORS

In this paper, we consider 3 objects–a red ball, a blue ball, and a yellow ball– in virtual environment as a toy, food, and a shelter respectively. Here are the behavior groups and their stimulus-reaction relationship.

- Play: Playing with a toy(red ball) when it is detected. Rity can 1) chase it when it's not near, 2)ride it when it's near, and 3) look around(search) when it disappers.
- Boredom: pestering or searching for a toy when it's not detected and magnitude of Boredom goes beyond a certain threshold.
- Food: Eating food(a yellow ball). Rity can 1) chaes it when it's not near, 2) recharge(eat food) when it's near, and 3) move backward after finishing recharge.
- Hunger: Complaining of hunger or searching for the yellow ball when it's not detected and magnitude of Hunger goes beyond a certain threshold..
- Comfort: Resting or sleeping near a blue ball. Rity can 1) chase it when it's not near, 2) rest when it's near and environment is bright enough, and 3) sleep when it's near and environment is dark.
- Fatigue: Yawning or searching for a blue ball when it's not detected and magnitude of fatigue goes beyond a certain threshold. Besides, even if the blue ball is detected Rity yawns when environment is too bright.

- Collision avoidance: Avoiding obstacles when moving around. Rith can 1) turn left or 2) turn right when obstacles are detected.
- Human interaction: Interacting with human(user). User affects Rity through touching it and giving voice command. Rity can 1) feel happy(hurrah, head up, etc) when user appears, praises it, or touches its head and body, 2) feel sad(hit ground, head down) when user disappears or scolds it, 3) feels angry(shake head, roar, etc) when user forbids its current behavior, hits its head and body, or shakes it greatly, 4)feel shy(hide face) when user shakes it slightly, and 5) achieve voice command(dance, stop, etc) from user.
- Direct reaction: Perceiving environmental changes and reacting to them. Rity can 1) feel fear(flinch, hide head, etc) when loud sound is made suddenly or it's getting dark suddenly, and 2) cover its eyes when it's getting bright suddenly.
- Wander: Strolling when there's no stimulus. Rity can walk and run left/right/forward.

2.3. BEHAVIOR SCENARIOS

With 3 pre-defined objects, we make bevaior scenarios of Rity as follows.

- Usually Rity wanders when a red ball is not detected. At this time if magnitude of Boredom goes beyond a certain threshold, it does Boredom behavior described above. When the red ball is found, Rity plays with it.
- If magnitude of Hunger goes beyond a certain threshold, Rity feels hungry. If a yellow ball is detected, Rity goes to the yellow ball and eats food (recharges). If not, it does Hunger behavior described above.
- If magnitude of Fatigue goes beyond a certain threshold, Rity feels tired. If a blue ball is detected, Rity gets close to the blue ball and rests or falls asleep depending on brightness of environment. In other cases, it does Fatigue behavior described above.
- Each motivation feature is associated with two behavior groups: Boredom with Play and Boredom, Hunger with Food and Hunger, and Fatigue with Comfort and Fatigue.
- Motivated behaviers and non-motivated behaviors are equal. For instance, even though Rity found and plays with the ball, if Rity senses other stimulus such as voice command, it prefers to react to those stimuli. A preference among motivation features is Fatigue>Hunger>Boredom. Therefore when feeling more than two motivation features at the same time, Rity acts according to the preference. A preference among non-motivated behaviors is Collision avoidance>Human interaction>Direct reaction>Wander.
- As for emotion features, they also have thresholds. The largest magnitude of feature dominates other features. If a behavior from stimuli is not consistent with current emotion feature, Rity doesn't react to those stimuli. For instance, if emotion feature of Rity is defined as Happy, it doesn't get angry and show its feeling although user hits its head.

3. IMPLEMENTATION

First we introduce a basis of Bayesian network needed to build the designed behavior controller from [4],[5] and build the controller.

3.1. BAYESIAN NETWORK

Solving problems applying probabilistic method becomes popular in many areas. Fuzzy inference and Bayesian network are fine examples. Bayesian network has the some advantages in that: 1) It is easy for to learn through simple probability computations, 2) Bayesian network can treat semantic symbol data, and 3) it is possible to design an adaptive model .

In order to explain Bayesian network, we must understand Directed Acyclic Graph (DAG) first. DAG consists of nodes and links between two adjacent nodes. Each node represents one random variable. We can calculate a conditional probability through a link. Fig. 3 shows an example of simple DAG.

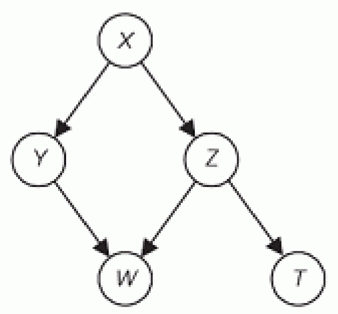


Fig. 3. An example of simple DAG

In Fig. 3, for each node, a node which gives an arrow is called "Parent" and a node which receives an arrow is called "Child". For example, a node Z receive an arrow from X and gives arrows to W and T. Therefore X is parent and W, T are children of node Z.

Theorem 1. Let a DAG G be given in which each node is a random variable and let a discrete conditional probability distribution of each node given values of its parents in G be specified. Then the product of these conditional distributions yields joint probability distribution P of the variables, and (G, P) satisfies the Markov condition.

Proof: See the [4].

Definition 1. If (G, P) satisfies the Markov condition, (G, P) is a Bayesian network.

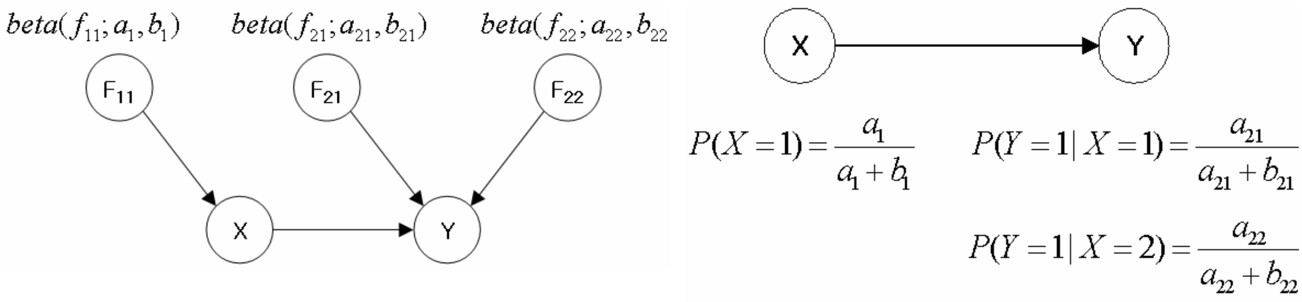
For example, in Fig. 3, we can get conditional probability distribution of each node given values of its parents as P(X), P(Y|X), P(Z|X), P(W|Y, Z), P(T|Z) and by theorem 1, can get joint probability distribution $P(X,Y,Z,W,T) = P(X) \times P(Y \mid X) \times P(Z \mid X) \times P(W \mid Y,Z) \times P(T \mid Z)$. Also, because (G, P) of Fig. 3 satisfies the Markov condition, by definition 1, (G, P) of Fig. 3 is Bayesian network.

3.2. PARAMETER LEARNING

There are two ways for learning Bayesian network: one is learning DAG, called "structure", from data and the other is learning conditional probability distributions, called "parameter", from pre-defined structure. The former is more complicated and more difficult to learn than the latter. Both learnings are explained well in [4]. In this paper, because we focus on parameters, we introduce parameter learning by taking an example.

Suppose a simple Bayesian network of two nodes X and Y, X→Y, is given. X, Y are binomial random variables. That is, values of each variables can be expressed as 1 and 2.

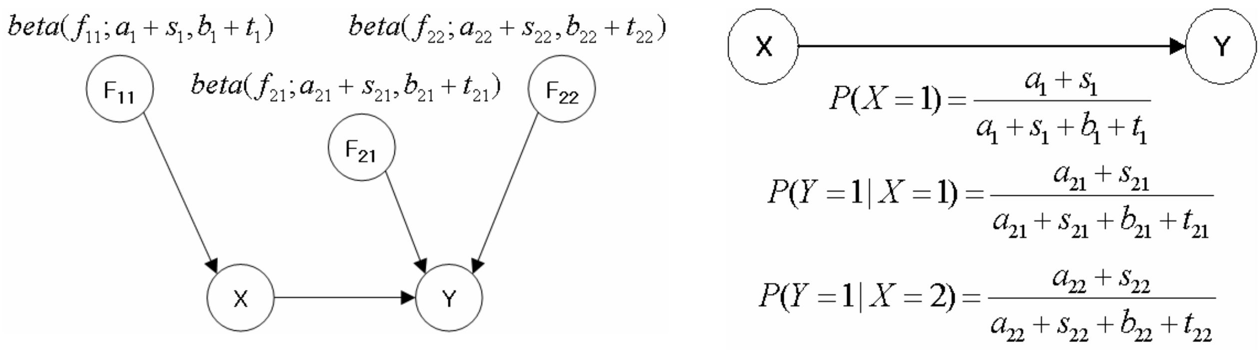
At first we initialize parameters. Consering empty parent node X, we can specify the number of times which value of X becomes 1 and 2 as a_1 and b_1. Also, considering Y whose parent is X, we can specify the number of times which value of Y becomes 1 and 2 when X is 1 and 2 as a_{21}, b_{21}, a_{22}, and b_{22} respectively. Then probability distributions of random variable F_{11}, F_{21}, and F_{22} become beta distribution $beta(f_{11};a_1,b_1)$, $beta(f_{21};a_{21},b_{21})$, and $beta(f_{22};a_{22},b_{22})$ respetively. Fig. 4(a) shows the augmented Bayesian network with calculated beta distributions. Finally values of parameters are calculated as Fig. 4(b).



(a) (b)

Fig. 4. A augmented Bayesian network of representing the prior probability distribution for the experiment is shown in (a); a Bayesian network contaning the prior marginal distribution of X and Y is shown in (b)

Next, we update parameters. According to experimental data, we can obtain the number of times which value of X becomes 1 and 2 as s_1 and t_1, the number of times which value of Y becomes 1 and 2 when X is 1 and 2 as s_{21}, t_{21}, s_{22}, and t_{22} respectively. Then probability distributions of random variable F_{11}, F_{21}, and F_{22} become beta distribution $beta(f_{11};a_1+s_1,b_1+t_1)$, $beta(f_{21};a_{21}+s_{21},b_{21}+t_{21})$, and $beta(f_{22};a_{22}+s_{22},b_{22}+t_{22})$ respetively. Fig. 5(a) shows the augmented Bayesian network with updated beta distributions. Finally updated values of parameters are calculated as Fig. 5(b).



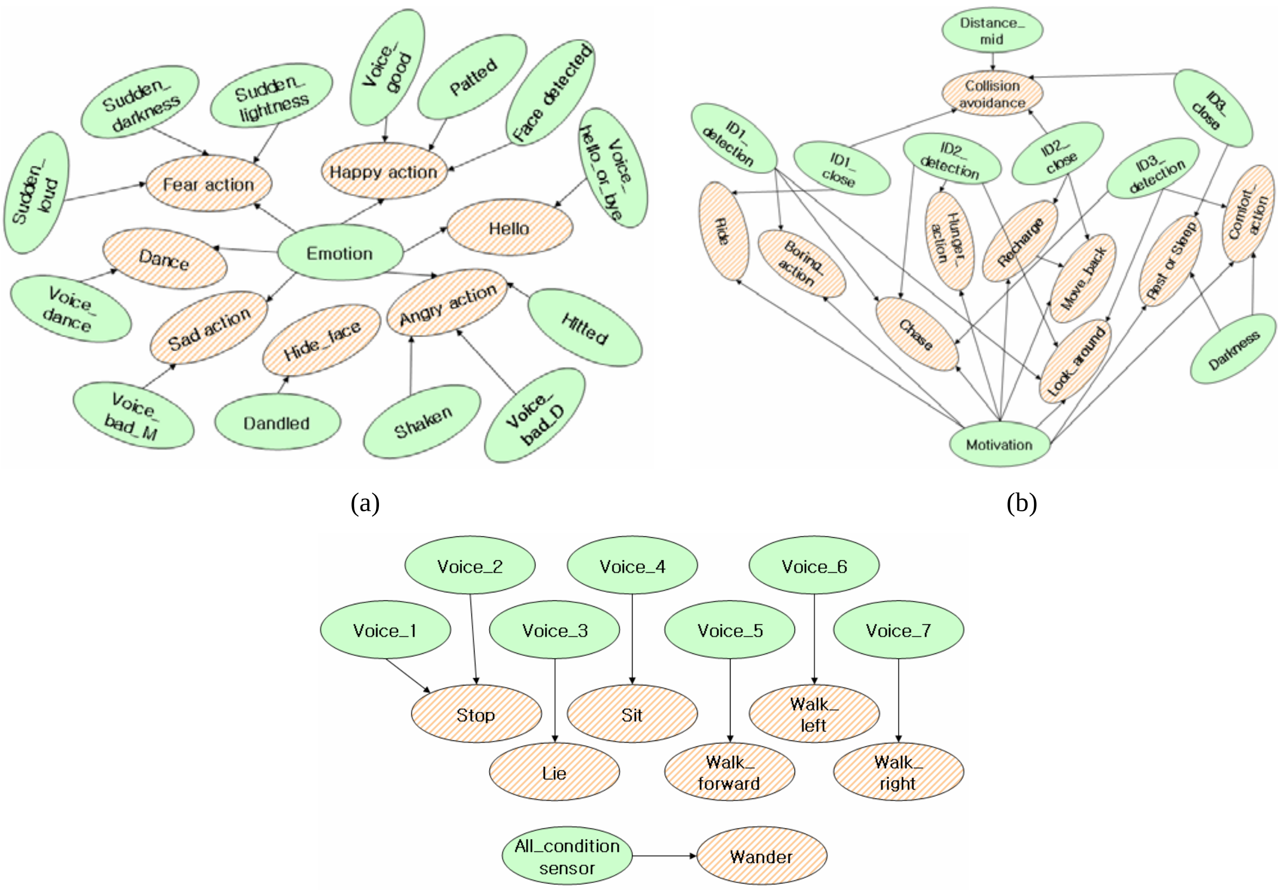
(a) (b)

Fig. 5. A augmented Bayesian network of containing the posterior probability distribution given data obtained by performing the experiment is shown in (a); a Bayesian network contaning the posterior marginal distribution of X and Y is shown in (b)

If random variable becomes multinomial, we can easily cacluate probability distributions by setting beta distributions as $beta(f_1,f_2,\ldots,f_{r-1};a_1,a_2,\ldots,a_r)$ and considering conditional probability distribution of each node given all values(they will be more than two) of its parents.

3.3. BEHAVIOR CONTROLLERS

We design a Bayesian network for the behavior controller according to behavior scenarios. This network consists of 24 result nodes representing behaviors/behavior groups and 37 conditional nodes representing stimuli such as sensor input, voice command, etc. Links between nodes can be obtained from stimulus-reaction relationship of behaviors and behavior scenarios. Since a whole network is very complicated and difficult to show at a time, we divide it into 3 sub-Bayesian networks: emotion-oriented, motivation-oriented, and else. Fig. shows the designed behavior controller.



(a) (b)

(c)

Fig. 6. A Bayesian network of behavior controller: (a) a sub-Bayesian network for emotion-oriented behaviors, (b) a sub-Bayesian network for motivation-oriented behaviors, and (c) a sub-Bayesian network for else behaviors

4. EXPERIMENTAL RESULTS

Since experimental results for all inputs and outputs is complicated and is difficult to describe at once, we show some parts of them whose inputs and outputs can indicate the characteristics of system mentioned above.

At first we observe the parameter(conditional probability distribution) changes during parameter learning stage. Here we observe the node “Stop”, “Voice command 1”, “Voice command 2”. As seen in Fig. 6(c), voice command 1 and 2 are parents of node “Stop”. So a node “Stop” has 4 beta distributions. Initally all parameter values are set at 0.5. Then, according to experimental data– Rity stops when user give voice command 1 or 2– , parameter values are varying. Fig. 7 shows the values of parameters depending on training generations. A probability that chosen behavior will become “Stop” is high when user give voice command 1 and 2. On the contrary, if voice command 1 or 2 isn’t given, the probaibility is low. Also, in reality, user can’t give voice command 1 and 2 simultaeously. Therefore in Fig. 7. we can show that P(Behavior=Stop|Voice command 1 is given, Voice command 2 isn’ given) and P(Behavior=Stop|Voice command 1 isn’t given, Voice command 2 isn’ given) increase and P(Behavior=Stop|Voice command 1 and 2 aren’t given) decreases as traning generation increases, and also P(Behavior=Stop|Voice command 1 and 2 are given) doesn’t change.

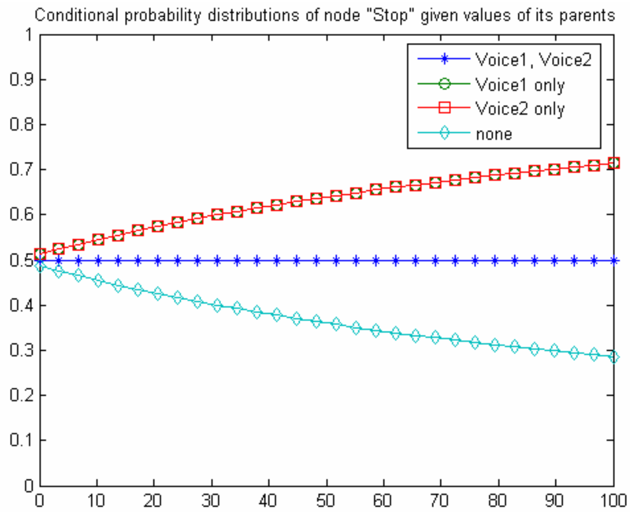
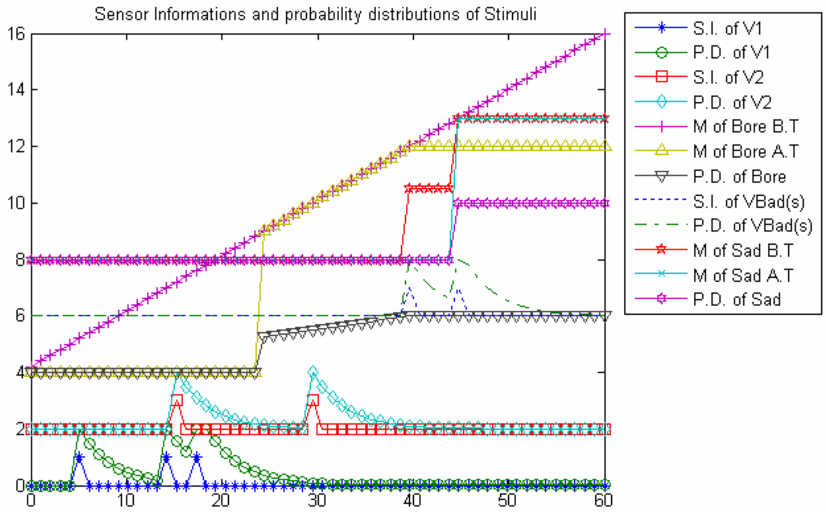


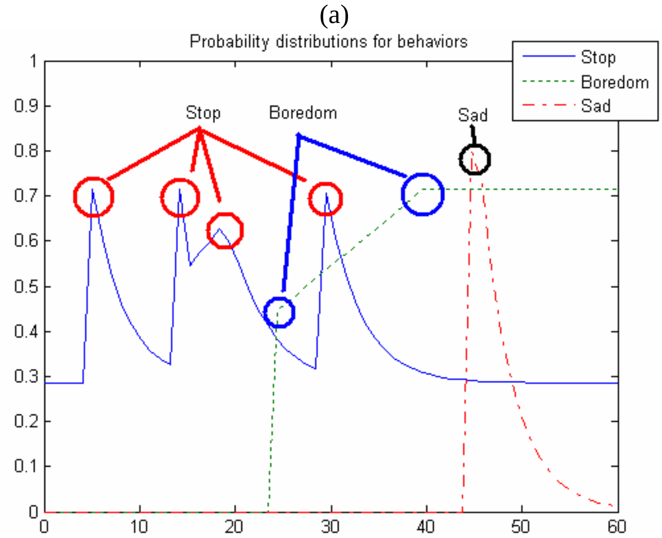
Fig. 7 Conditional probability distributions of node “Stop” given values of its parents depending on training generation

Next we apply inputs according to 3 nodes(behaviors/behavior groups) included in each sub-Bayesian network described in Fig. 6. We define sample behaviors as “stop” from a sub-Bayesian network for else behaviors, “Boredom” from a sub-Bayesian network for motivation-oriented behaviors, and “sad” behaviors from a sub-Bayesian network for emotion-oriented behaviors. As shown in Fig. 8, from 0 to 14 generations only voice command 1 related to behavior “stop” is given and a probability distribution of behavior “stop” is much higher than others. Therefore behavior “stop” is chosen. From 15 to 19 generations voice command 1 and 2 related to behavior “stop” are given and a probability distribution of behavior “stop” is still higher than others. Therefore behavior “stop” is chosen, too. Up to 19 generations, even though magnitude of Boredom increases continuously, it doesn’t go beyond the threshold. Therefore Boredom feature is not considered. From 20 to 30 generations there is no sensor inputs and magnitude of Boredom goes beyond the threshold. Therefore a probability distribution of behavior group “Boredom” is higher than others and behavior group “Boredom” is chosen. From 30 to 44 generations, sensor input voice_bad related to “sad” behaviors is given. However magnitude of Sad doesn’t go beyond the threshold. Therefore Sad feature is not considered and behavior group “Boredom” is chosen since a probability distribution of behavior group “Boredom” is higher than others. From 45 to 60 generations, voice_bad related to “sad” behaviors is given too. At this time magnitude of Sad goes beyond the threshold. Therefore a probability distribution of “sad” behaviors is higher than others and one of “sad” behaviors is chosen.

In view of the results so far achieved, we can think that designed behavior controller decides right behavior according to stimulus-reaction relationship and behavior scenarios.



(a)



(b)

Fig. 8 Probablity distributions and magnitude of features in internal status system: (a) sensor data and probability distributions of conditional nodes(stimuli) and magnitude of features in internal status system before and after passing thresholds, (b) probability distributions for behaviors

5. CONCLUSTION

In this paper, based on pre-defined behavior scenarios, we build a behavior controller using Bayesian network, and by using the designed controller, we can make probabilistic decisions. Through experimental results, we can show that control probability distributions of each node(behavior/behavior group) given values of its parents changes depending on stimuli(sensor inputs) and value of probability vector follows behavior scenarios well.

For future work, not only updating control parameter distributions of pre-defined Bayesian network, we can also define conditional nodes and result nodes, and built Bayesian network after getting enough experimental data.

6. ACKNOWLEDGMENT

This work is part of project Development of real time monitoring the welding Gap/Profile.

REFERENCES

[1] TETSUNARI IMAMURA, *User Adaptation of Human-Robot Interaction Model based on Bayesian Network and Introspection of Interaction Experience*, IROS Proceedings. 2000 IEEE/RSJ International Conference, Vol. 3(2000), pp.2139 - 2144.

[2] YOUNG CHEOL GO., *Probabilistic approach to model intelligent robot services*, ICACT The 8th International Conference, Vol.2(2006), pp.1221-1224.

[3] HYEUN JEONG MIN, *Navigation of a mobile robot using behavior network with Bayesian inference*, IEEE International Conference on Mechatronics and Automation, Vol.3(2005).

[4] RICHARD N., *Learning Bayesian Networks*, Prentice Hall, 2003.

[5] FINN J., *Bayesian Networks and Decision Graphs*, Springer-Verlag, New York 2001.

[6] Artificial Creatures Rity(2D), RIT Lab, KAIST, http://rit.kaist.ac.kr

[7] Yong-duk Kim, Implementation of artificial creature and interactive learning using behavior set, Master's Thesis, KAIST, 2002