LEARNING BEHAVIOR SELECTION THROUGH INTERACTION BASED ON EMOTIONALLY GROUNDED SYMBOL CONCEPT

TSUTOMU SAWADA
Information Technologies Laboratories, Sony Corporation,
6-7-35 Kitashinagawa Shinagawa-ku, Tokyo, 141-0001, Japan
tsawada@pdp.crl.sony.co.jp

TSUYOSHI TAKAGI
Entertainment Robot Company, Sony Corporation,
5-11-3 Shinbashi Minato-ku, Tokyo, 105-0004, Japan
takagi@erc.sony.co.jp

YUKIKO HOSHINO
Life Dynamics Laboratory Preparatory Office, Sony Corporation,
6-7-35 Kitashinagawa Shinagawa-ku, Tokyo, 141-0001, Japan
yukiko@pdp.crl.sony.co.jp

MASAHIRO FUJITA
Information Technologies Laboratories, Sony Corporation,
6-7-35 Kitashinagawa Shinagawa-ku, Tokyo, 141-0001, Japan
mfujita@pdp.crl.sony.co.jp

In this paper, we propose a learning algorithm for action selection mechanism in the EGO architecture, which we proposed for autonomous behavior control of a humanoid robot. The concept of behavior value is introduced for action selection. The behavior value of each behavior module depends on external stimuli and internal states, and the behavior module with the higher behavior value is selected in the situation. We address the importance of learning the behavior value of each behavior. We describe how to compute behavior values for behavior modules through interaction with humans and environment. We implemented the learning algorithm on QRIO SDR-4X II, a small humanoid robot, and confirmed that for a given interaction driven behavior module, a high behavior value is obtained when interacting with a friendly user. The same tendency is obtained for a proper color painted ball for soccer play behavior module.

Keywords: EGO architecture, behavior value, learning, QRIO SDR-4X II.

1. Introduction

We have been proposing autonomous behavior control architecture, named EGO Architecture, for consumer entertainment applications[1]. For the purpose, we developed a small humanoid robot QRIO SDR-4X (later QRIO). It is necessary for such a robot to walk around in home environment, to respond to social cues and other stimuli, to find and identify users, and to communicate with users naturally. There are many technologies, such as real-time dynamic walking control, map-building of environment, human detection and identification, speech recognition and synthesis, and natural language processing for verbal communication. Moreover, it is important to behave spontaneously and naturally.

The EGO architecture is developed to integrate the technologies and to make QRIO behave spontaneously and naturally.

From Behavior Control Architecture point of view, how to coordinate behaviors properly is one of the important issues. In the Behavior Based architecture[1], it is described that the "releasers" of behaviors coordinate behaviors, and the releasers are carefully designed and debugged by human. Usually, the releasers are described as a TRUE-FALSE logic table, and one TRUE releaser releases the corresponding behavior in the situation[2].

In our EGO architecture, we assign "behavior value" for each behavior module, and proper behaviors are coordinated based on that value. The behavior value could be considered as Q-value in reinforcement learning, where the action with the higher Q-value is selected to get the higher reward. In the similar way, the behavior with the higher behavior value is selected to regulate the internal variables. The details will be described in the later section, but in short the internal variables must be regulated in certain ranges. This is a key for autonomous or spontaneous behavior of the EGO architecture.

Let us return to the action selection issue, as the releaser is programmed manually, the behavior value is usually programmed or assigned manually. However, it is clear that when the system is getting complex as behaviors and target objects are increased, it is difficult to determine these behavior values manually. Moreover, in some cases it is impossible to determine the behavior values before the robot actually interacts with targets. For example, if there are a user (USER-A) who likes to interact with the robot, and a user (USER-B) who doesn't like to interact with the robot, the robot has to determine the behavior value of interaction behavior module in such a way that a higher behavior value for USER-A and a lower behavior value for USER-B are set. These values can not be assigned before the robot interacts with users.

We already proposed Emotionally Grounded Symbol concept[3,4], in which symbols are grounded to emotional system, which corresponds to the system with the internal variables. The learning algorithm of the behavior value is an example of implementation of the Emotionally Grounded Symbol concept.

In this paper, first we describe an overview of the EGO architecture, followed by how to compute behavior values, and how to coordinate behaviors properly based on the behavior values. Then, we describe how to learn the behavior values through interactions with target objects, some of which are initially set by programmer, and some of which are new targets. We implemented the learning mechanism in the EGO architecture and performed some feasibility studies. We describe several implementations and presents results of experiments using QRIO. Then, comparing with existing architectures, we discuss some features of the EGO architecture with respect to our learning mechanism.

Note that regarding the terminologies in this paper, because the EGO architecture is inspired by Ethological studies[5], we often use terminologies for animals' behaviors for our robot behaviors such as EAT for an energy charge behavior and NURISHMENT for an internal variable corresponding to battery energy. In EGO

architecture, if a robot is HUNGRY then EAT, a battery charging behavior, should have the higher behavior value.
One more thing we should note here is about the terminology of "action selection" and "behavior selection". The two terminologies are used in many fields in almost the same meaning. Because "action" sounds more primitive meaning than "behavior", we basically use behavior selection for our EGO architecture. However, when we refer to other articles, we try to use the original terminology in the literatures.

2. EGO ARCHITECTURE OVERVIEW

In this section, the individual software components of the EGO Architecture are briefly explained. Fig. 1 provides an overview. Please refer to the paper for more details in the EGO Architecture[6].

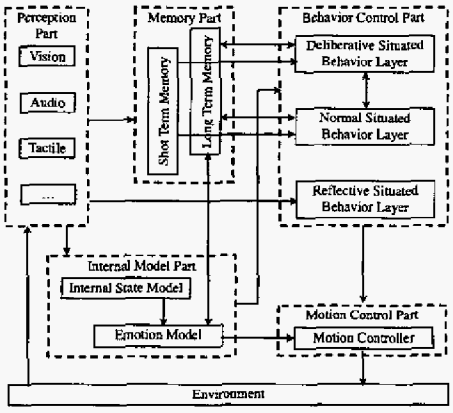


Fig. 1. Overview of the EGO Architecture

2.1. *Short Term Memory (STM)*

STM integrates the results of perception. From audio perception, STM receives the result of not only speech recognition but also sound source direction by multi microphone localization. As for vision perception, STM can obtain the result of face recognition with its associated direction and distance computed from stereo vision. In the case that both, audio and visual directions, are same, STM merges the results to indicate that they are from the same user.
STM can also compute relative positions to detected objects (face and ball etc.) through kinematics. Therefore STM can store and recall results located outside of the limited view range.

2.2. *Long term memory (LTM)*

LTM associates the recognition results with an internal state. For example, LTM can associate an acquired name with an identified object or an identified voice, and change the internal state associated with a target object. Details of LTM are described in the paper[8].

2.3. *Internal state model (ISM)*

ISM maintains various internal state variables. It alters their values depending on the passage of time and incoming external stimuli. Basically, a behavior module is selected in order to keep these internal state variables within proper ranges. ISM is the core for spontaneous behavior and response generation to external stimuli.

2.4. *Emotion model (EM)*

EM has 6+1 emotions, which are ANGER, DISGUST, FEAR, JOY, SADNESS, SURPRISE, and NEUTRAL, based on Ekmann's proposal[7]. Each emotion has an associated value. They are determined based on self-preservation. The determination of self-preservation is composed of self-crisis and self-crisis expectation. The value of self-crisis is evaluated from external stimuli. Detail of this evaluation is described in the paper[8].

2.5. *Situated behavior layer (SBL)*

SBL controls behavior modules. Each behavior module has two basic functions, monitor and action.

Monitor function periodically and concurrently creates a value, which is called Behavior Value (*BV*), using internal state variables and external stimuli. It indicates how relevant the behavior is for the situation (e.g., observing an object and a sound event etc.). The details of this computation are described below.

A behavior module is selected by competition on the *BV*s. Greedy or soft-max is used as a selection policy. Then the selected behavior module is given execution permission.

Availability of necessary resources for execution, e.g., head, arm, speaker, etc., are also considered in the competition. In the case where there is no resource conflict among behavior modules, all of them are given execution permission and then execute concurrently.

After a behavior module is given permission, the action function executes the

behavior implemented as a state machine. Each node can output e.g. a motion command (designed motion command, walk command, and tracking command etc.) and can decide to state transition depending upon the given situation.
Figure 2 shows a behavior module and associated process.
A tree structure is used to organize the behavior modules. An abstract behavior can be divided into concrete and multiple sub-behaviors. For example, as shown in Fig. 3, "Soccer" can be decomposed into "Search ball", "Approach ball" and "Kick ball", also "Approach ball" can be decomposed into "Go to ball by walk", "Track ball by head", and "Speak for approach" etc.
In the parent behavior module in the tree structure, a monitor function can also determine the *BV* through the child *BV*s instead of evaluation through the internal state variables and external stimuli. The action function can also select a child behavior module instead of a motion command

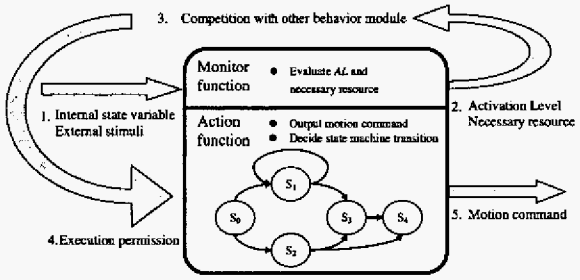


Fig. 2. Behavior module and process

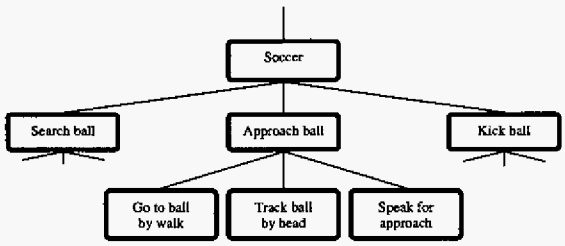


Fig. 3. Tree structure of behavior modules

SBL is organized in 3 modules, D-SBL (Deliberative SBL), N-SBL (Normal SBL) and R-SBL (Reflexive SBL). D-SBL realizes behavior control for deliberative behavior, N-SBL realizes behavior control for homeostatic behavior, and R-SBL realizes behavior control for quick responses.
Please refer to the proceedings for more details on SBL[6,9].

3. LEARNING BEHAVIOR VALUE

In this paper, we focus on the learning of *BV's* to realize homeostatic behavior in N-SBL. Performing a behavior on a target would cause changes in the internal state variables. Each behavior module evaluates how much the internal state would change as a result of performing the behavior. The association of it is learned in each behavior module. Evaluation and learning of *BV* are described in detail in the following subsection.

3.1. *Evaluation of behavior value*

Each *BV* is composed of a *Motivation* value (*Mot*) and a *Releasing* value (*Rel*).
The evaluation of *Mot*, *Rel* and *BV* is described using the following example behavior "Approach a target object for eating it". EGO architecture is based on ethological study. The example is for an agent to regulate the NOURISHMENT state variable. From the viewpoint of robotics, NOURISHMENT is interpreted as charge of battery, eating as pseudo-eating, that is charging-battery, and object as battery station.
The motivation value is the degree to which the instinct drives the behavior module. It is derived from internal state variables and is composed of instinct values.
An instinct value ($Ins[i]$) is designed for each specific internal state variable ($Int[i]$).
Two examples for NOURISHMENT and FATIGUE are shown in Fig. 4 (a), (b) and can be interpreted as follows. The less nourishment there is, the larger the instinct to eat it is. Also, in the case of large nourishment, this instinct turns negative to realize a moderation or reduction in eating behavior. Fatigue has a negative effect. The more fatigue there is, the less the value of the instinct associated with it.
Mot is evaluated as shown in Eq. (1).

$$Mot = \sum W_{Mot}[i] \cdot Ins[i] \tag{1}$$

where $W_{Mot}[i]$: Weight of $Ins[i]$

The releasing value is the degree regarding how much an external stimuli would satisfy an internal state as a result of the behavior. It is derived from an internal state variable and the external stimuli. It is composed of a satisfaction value and the expectation of satisfaction value.
A satisfaction value ($Sat[i]$) is designed for each specific internal state valuable. Examples for NOURISHMENT and FATIGUE are shown in Fig. 4 (c), (d).
To evaluate the expectation of satisfaction value ($ESat[i]$), the behavior module maintains a database on expectation of change in the internal state variable ($dInt[i]$)

against the result of the behavior for the given external stimuli.

Figure 5 is an example where the behavior module expects a change in NOURISHMENT and FATIGUE when an external stimuli (OBJECT_ID, OBJECT_SIZE, and OBJECT_DISTANCE) is obtained. It means that when a target object is found which has OBJECT_ID = 1, OBJECT_SIZE = 100, and OBJECT_DISTANCE = 2000, NOURISHMENT would increase 20 and FATIGUE would increase 20 after approaching and eating the target object.

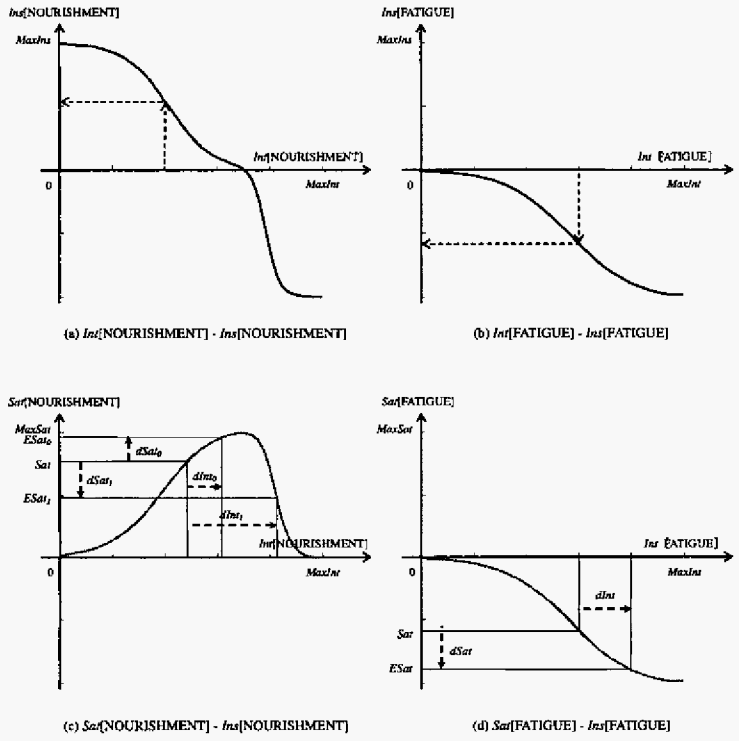


Fig. 4. *Ins*[*i*] - *Int*[*i*] and *Sat*[*i*] - *Int*[*i*] in the behavior module

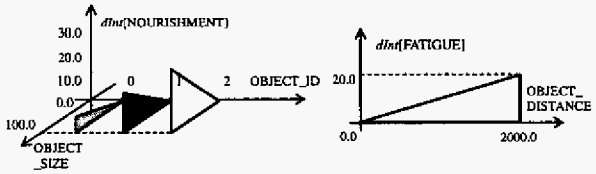


Fig. 5. Database about expectation of change in the internal state variable

ESat[*i*] and expectation of change in satisfaction value (*dSat*[*i*]) are shown in Fig. 4 (c), (d). They are interpreted as follows. When $dInt_0$ is determined by observing an $object_0$, the *dSat*[NOURISHMENT] is expected as positive. On the contrary, when $dInt_1$ is determined when observing another $object_1$, for example whose size is larger than $object_0$, the *dSat*[NOURISHMENT] is expected as negative due to overeating. *dInt* for fatigue is related to the distance of an observed object. The farther the distance is, the more dissatisfaction the agent receives.
Rel is evaluated by Eq. (2).

$$Rel = \sum W_{Rel}[i] \cdot \left(W_{dSat} dSat[i] + \left(1 - W_{dSat}\right) ESat[i]\right) \tag{2}$$

where $W_{Rel}[i]$: Weight of $(W_{dSat}dSat[i]+(1-W_{dSat})ESat[i])$
W_{dSat}: Weight of $dSat[i]$ against $ESat[i]$

BV is evaluated from *Mot* and *Rel* by Eq. (3).

$$BV = W_{Mot} Mot + \left(1 - W_{Mot}\right) Rel \tag{3}$$

where W_{Mot}: Weight of *Mot* against *Rel*

Note that when there is no external stimuli for the behavior module, *BV* is set to 0, so that behavior module is never selected.

3.2. *Learning of change in the internal state variable*

As mentioned in the introduction, it is difficult to set *BV* properly. And it is important that *BV* changes properly through interactions with external stimuli.
In the evaluation of *BV*, each behavior module expects *dInt*[*i*] based on the database through external stimuli. And as a result of the behavior, internal state variable really changes. In this paper, *dInt*[*i*] is renewed by feedback of real change in internal state variable and parameters of *BV* are learned.
Figure 6 shows the process of the learning using an example of "eat a target object". The behavior module evaluates *BV* from *Int*[NOURSHMENT] and external stimuli OBJECT_ID = 2, OBJECT_SIZE = 100.0 in the database.

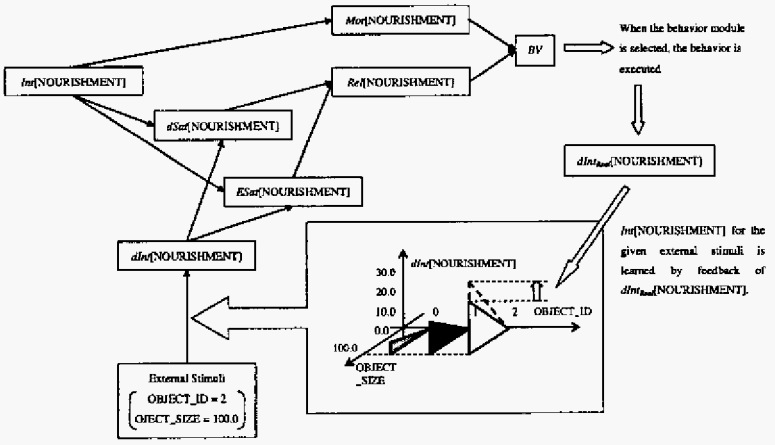


Fig. 6. Process of the learning

Execution of the behavior (eat the target object) causes real change in NORISHMENT ($dInt_{Real}$[NORISHMENT]). And $dInt$[NOURISHMENT] for the given external stimuli is learned by feedback of $dInt_{Real}$[NOURISHMENT] by following Eq. (4).

$$dInt[i] \Leftarrow (1-\alpha)dInt[i] + \alpha \cdot dInt_{Real}[i] \tag{4}$$

where α : Learning ratio

For unknown target object, a default $dInt[i]$ is set heuristically. Even if the default $dInt[i]$ is not proper at first, it would be learned properly because $dInt[i]$ grounds on real change in internal state variable through the process.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Let us consider two example behaviors (application) to proceed in the discussion. The first behavior, "Kick a ball" satisfies e.g. VITALITY. The second behavior, "Interact with an user" satisfies e.g. INTERACTION. Experiments are conducted to learn parameters of BV through QRIO interacting with faces and balls, and to behave autonomously based on learned BV in the real environment.

4.1. *Hardware component of QRIO*

Figure 7 shows QRIO's appearance. It is 580 [mm] height, approximately 7 [kg] with battery and having 38 DOF. It is a stand-alone robot with three CPUs. The first is for audio recognition and text-to-speech synthesis. The second is for visual recognition, short- and long-term memory, and the behavior control architecture. The third is dedicated to motion control. Remote processing power and robot control is also available through wireless LAN.

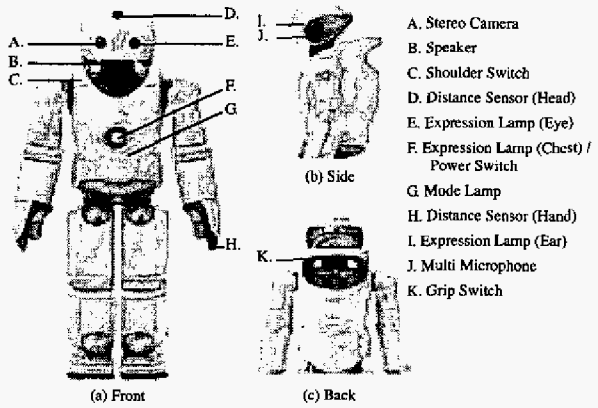


Fig. 7. Appearance of QRIO

4.2. *Implementation of experiment*

The tree structure of behavior modules for the application is shown in Fig. 8.
Soccer (*Sc*) sub tree has 3 child behavior modules *Soccer Search* (*ScSr*), *Soccer Approach* (*ScAp*), and *Soccer Do* (*ScDo*). They evaluate *BV* based on an internal state variable and external stimuli. *BV* of *Sc* is the maximum *BV* among its children.
ScAp focuses on VITALITY and FATIGUE as internal state variables, and BALL_ID and BALL_DISTANCE as external stimuli. BALL_ID = 0 means a red ball with radius 75 [mm] and weight 330 [g]. BALL_ID = 1 means a green ball with radius 75 [mm] and weight 110 [g].
BV is composed of *Mot* and *Rel* with $W_{Mot} = 0.4$.
Mot is composed of *Ins*[VITALITY] and *Ins*[FATIGUE], which are shown in Fig. 9 (a) and (b), with W_{Mot}[VITALITY] = 0.8 and W_{Mot}[FATIGUE] = 0.2.
Rel is composed of *dSat*[VITALITY], *dSat*[FATIGUE], *ESat*[VITALITY] and *ESat*[FATIGUE], which are shown in Fig.9 (d) and (e), with W_{Rel}[VITALITY] = 0.8, W_{Rel}[FATIGUE] = 0.2 and $W_{dSat} = 0.0$.
dInt[VITALITY] and *dInt*[FATIGUE] are estimated from BALL_ID and BALL_DISTANCE. Default value for them are shown in Fig. 10 (a) and (b).

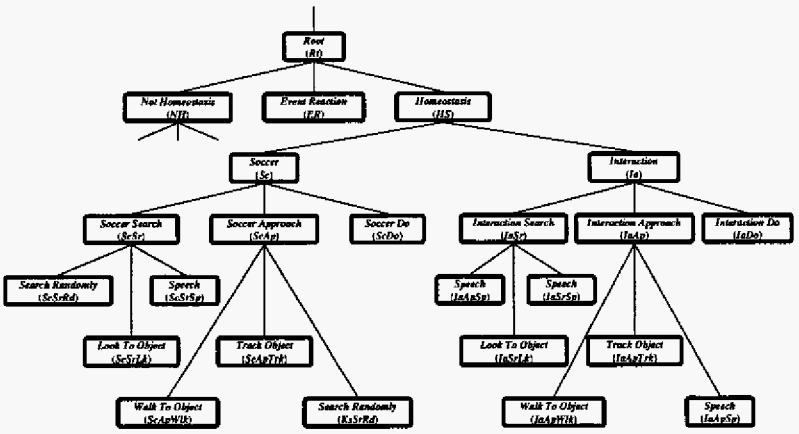


Fig. 8. Tree structure for the application

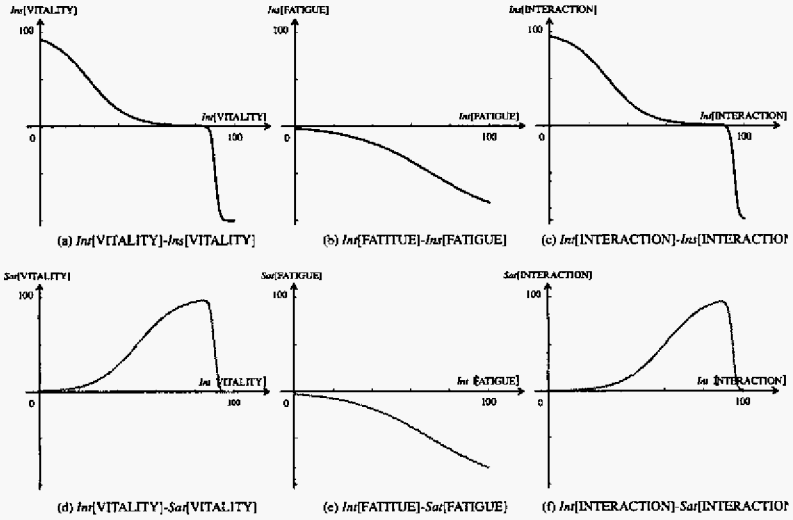


Fig. 9. *Ins*[*i*] against *Int*[*i*] and *Sat*[*i*] against *Int*[*i*]

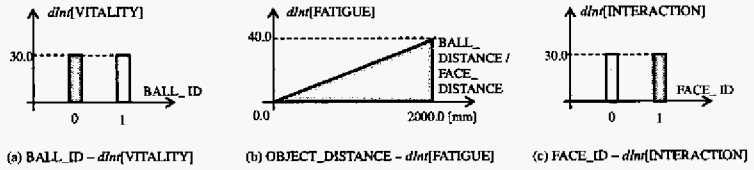


Fig. 10. Default *dInt*[*i*] against external stimuli

ScSr focuses only on VITALITY as an internal state variable. It does not focus on external stimuli. Evaluation of *BV* is the same as for *ScAp* except for the values on FATIGUE, which are set to 0.

ScDo focuses only on VITALITY as an internal state variable and BALL_ID as an external stimuli. Evaluation of *BV* is same as *ScAp* except for the values on FATIGUE, which are set to 0. On the condition that the ball distance is not in the proper range for kick motion (0 - 400 [mm]), *BV* = 0. Note that the distance is not used to evaluate *Rel*.

Motion commands for search and approach a ball are output in children behavior modules of *ScSr* and *ScAp*. *ScDo* outputs kick motion commands and $dInt_{Real}$[VITALITY] is estimated from the distance to the ball after kicking it in the action function. The estimation is defined by Eq. (5). It means that *Int*[VITALITY] increases by 50 when distance of kicked ball is 1000 [mm].

$$dInt_{\text{Real}}[\text{VITALITY}] = 0.05 * BDst \tag{5}$$

where *BDst*: Distance to the ball [mm]

Interaction (*Ia*) sub tree, composed of *Interaction Search* (*IaSr*), *Interaction Approach* (*IaAp*) and *Interaction Do* (*IaDo*), has the same structure as *Sc* except for internal state variable and external stimuli. INTERACTION and FACE_ID is specified instead of VITALITY and BALL_ID respectively.

In action function of *IaDo*, QRIO requests interaction with the user at first state machine node. When the face becomes much closer, interaction motion command is executed and *Int*[VITALITY] increases in 50 (that is $dInt_{Real}$[VITALITY] = 50). On the other hand, if it does not become much closer for a while, QRIO gives up interaction and the state machine is finished. In this case, *Int*[VITALITY] does not increase (that is $dInt_{Real}$[VITALITY] = 0).

Ins[INTERACTION], *Sat*[INTERACTION] and default of *dInt*[INTERACTION] are shown in Fig. 9 (c), (f) and Fig. 10 (c) respectively.

On the condition that the distance to the detected face is not in the proper range for interaction (100 - 500 [mm]), *BV* = 0.

For the implementation of learning, we focus on *dInt*[VITALITY] and *dInt*[INTERACTION] against each target object BALL_ID and FACE_ID. And they are same instance in *ScAp* and *ScDo*, *IaAp* and *IaDo* for each.

Learning ratio α is set to 0.4.

Not Homeostasis (*NH*) is not a homeostasis behavior module, so *BV* = 10 constantly. It outputs an idle motion command like leaning the head to one side, tracking a face, etc. When *BV* of all homeostatic behavior modules are low (all internal states are satisfied), *NH* is executed.

Event Reaction (*ER*) does not output any motion command by itself. When an event triggering a reflexive behavior comes, *ER* reserves required resources by setting *BV* = 100 to prevent a homeostatic behavior module from executing and

interfering with the reflexive behavior. A parent behavior module selects its child behavior modules using a greedy policy based on the children's *BV*.
Figure 11 shows the appearances of the experiment.

(a) Kick a ball (b) Interact with a person

Fig. 11. Appearance of the experiment

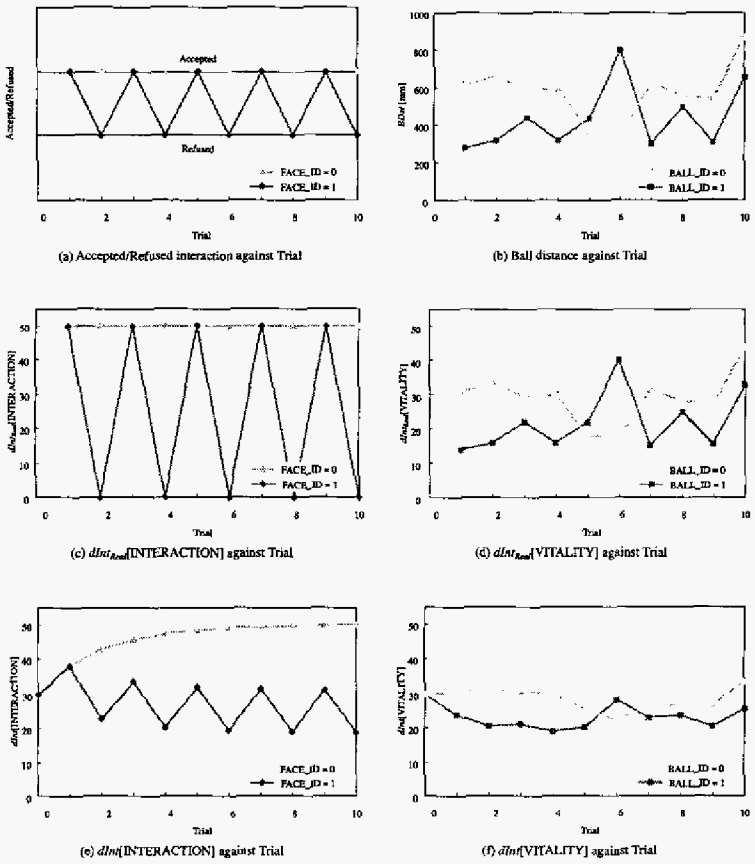
4.3. *Experiment of learning behavior value*

In the condition that *Int*[VITALITY] = 80, *Int*[FATIGUE] = 10, *Int*[INTERACTION] = 20, *Ia* sub tree is active and *Sc* sub tree never be active because *BV* of *ScSr*, *ScAp*, and *ScDo* are negative at all times. QRIO tries to search a face, approach to the user, and request interaction with the user. It is executed 10 times for each FACE_ID = 0, 1.
Figure 12 (a), (c) and (e) show the experimental results of learning *dInt*[INTERACTION].
User, whose FACE_ID = 0, accepts the request of interaction always. On the contrary, User, whose FACE_ID = 1, accepts it every other time. Then $dInt_{Real}$[INTERACTION] = 50.0 is obtained successively for FACE_ID = 0, and $dInt_{Real}$[INTERACTION] = 0.0 is obtained every after obtaining $dInt_{Real}$[INTERACTION] = 50.0 for FACE_ID = 1. (See Fig. 12 (a) and (c))
As a result of the learning, *dInt*[INTERACTION] for FACE_ID = 0 gradually converges to $dInt_{Real}$[INTERACTION] = 50.0 and becomes *dInt*[INTERACTION] = 49.9. On the contrary *dInt*[INTERACTION] for FACE_ID = 1 becomes 18.8 with oscillation. (See Fig. 12 (e))
In the condition that *Int*[VITALITY] = 20, *Int*[FATIGUE] = 10, *Int*[INTERACTION] = 80, *Sc* sub tree is active and *Ia* sub tree never be active because *BV* of *IaSr*, *IaAp*, and *IaDo* are negative at all times. QRIO tries to search a ball, approach to the ball, and kick the ball. It is also executed 10 times for each BALL_ID = 0, 1.
Figure 12 (b), (d) and (f) show the experimental results of learning *dInt*[VITALITY].
In the result of ball distance, the average is 436.1 [mm] for BALL_ID = 0 and

577.9 [mm] for BALL_ID = 1. It would be caused of difference of ball weight (330 [g] for BALL_ID = 0, 110 [g] for BALL_ID = 1). Because green ball is righter than red ball, it goes further when it is kicked.

They are less consistent (Standard deviation σ =174.3 [mm] for BALL_ID = 0 and σ =148.1 [mm] for BALL_ID = 1). It would be caused of interaction with real environment, that is ball recognition error, kick motion error, friction of the floor, etc. (See Fig. 12 (b)) And $dInt_{Real}$[VITALITY] is obtained like Fig. 12 (d) for each BALL_ID.

As a result of learning, *dInt*[VITALITY] for BALL_ID = 0 becomes 25.4 and BALL_ID = 1 becomes 33.6 as shown in Fig. 12.



(a) Accepted/Refused interaction against Trial
(b) Ball distance against Trial
(c) $dInt_{Real}$[INTERACTION] against Trial
(d) $dInt_{Real}$[VITALITY] against Trial
(e) *dInt*[INTERACTION] against Trial
(f) *dInt*[VITALITY] against Trial

Fig. 12. Experimental result of learning *dInt*

4.4. *Experiment of autonomous behavior based on learned behavior value*

Figure 13 (a) shows the experimental result of change in *BV* in the condition that *Int*[VITALITY] = 20, *Int*[FATIGUE] = 10, *Int*[INTERACTION] = 80 at $t = 0.0$ [s] with learned *dInt*[VITALITY], *dInt*[INTERACTION] in previous subsection, which *dInt*[VITALITY] = 25.4 for BALL_ID = 0, *dInt*[VITALITY] = 33.6 for BALL_ID = 1, *dInt*[INTERACTION] = 18.8 for FACE_ID = 0, and *dInt*[VITALITY] = 49.9 for FACE_ID = 1.

At first QRIO searches randomly for a ball in *BV*[*ScSr*] = 20. QRIO finds the ball with BALL_ID = 1 (green ball) at $t = 48.0$ [s], then *BV*[*ScAp*] increases to 44.5 and starts to approach the ball. QRIO reaches the distance where kick motion is effective at $t = 73.5$ [s], and then *BV*[*ScDo*] increases to 45.9 and kicks the ball.

After kicking the ball, *Int*[VITALITY] is satisfied to a level of 45.9 from 20 at $t =$ 99.0 [s] because ball distance is 519.0 [mm] and $dInt_{Real}$[VITALITY] = 25.9.

Int[VITALITY] is not satisfied enough in this condition. Then QRIO approaches and kicks the ball again from $t = 107.5$ [s], $t = 124.0$ [s] for each. Finally *Int*[VITALITY] is fully satisfied to a level of 45.9 from 75.8 at $t = 146.0$ [s] because ball distance is 577.6 [mm] and $dInt_{Real}$[VITALITY] = 28.9, and *NH* is executed by *BV*[*NH*] = 10 after $t = 153.0$ [s].

Note that *dInt*[VITALITY] is renewed online from 33.6 to 30.5 after first kick and from 30.5 to 29.9 in second kick.

Figure 13 (b) shows the experimental result in another condition that *Int*[VITALITY] = 20, *Int*[FATIGUE] = 10, *Int*[INTERACTION] = 20 at $t = 0.0$ [s]. User, whose FACE_ID = 1, claps his hands to make QRIO notice him during approach to the ball in *BV*[*ScAp*] = 34.5. And QRIO detects the sound at $t = 29.5$ [s] Then *ScAp* is interrupted and *ER* is executed in *BV*[*ER*] = 100 from $t = 29.5$ [s] to 45.0 [s]. The behavior module in R-SBL outputs a motion command to turn toward the sound source direction. At $t = 43.5$ [s] QRIO finds a face whose FACE_ID = 1, then *BV*[*IaAp*] increases to 26.6. Because *BV*[*ScAp*] is still larger than *BV*[*IaAp*], QRIO ignores the face at once and resumes the approach from $t =$ 45.5 [s] and kicks the ball after $t = 71.0$ [s]. Then QRIO searches, approaches, and requests interaction with the user from $t = 104.0$ [s], $t = 120.0$ [s], and $t = 143.0$ [s] respectively.

Figure 13 (c) shows the experimental result in same condition as previous experiment, that *Int*[VITALITY] = 20, *Int*[FATIGUE] = 10, *Int*[INTERACTION] = 20 at $t = 0.0$ [s]. User, whose FACE_ID = 0, claps his hands to make QRIO notice him during approach to the ball in *BV*[*ScAp*] = 34.5. And QRIO detects the sound at $t = 45.5$ [s]. Then *ScAp* is interrupted and *ER* is executed in *BV*[*ER*] = 100 from t = 45.5 [s] to $t = 83.0$ [s]. The behavior module in R-SBL outputs a motion command to turn toward the sound source direction. At $t = 73.5$ [s] QRIO finds a face whose FACE_ID = 1, then *BV*[*IaAp*] increases to 56.6. Now it is larger than *BV*[*ScAp*]. QRIO approaches the user, suspending its previous approach to the ball, and requests interaction with him at $t = 101.5$ [s]. After the interaction, *Int*[INTERACTION] is satisfied enough (*Int*[INTERACTION] = 70), and *BV* of *IaSr*, *IaAp*, and *IaDo* turn negative. QRIO restarts looking toward, approaching

and trying to kick it from $t = 136.0$ [s], $t = 187.5$ [s], $t = 205.0$ [s] respectively.

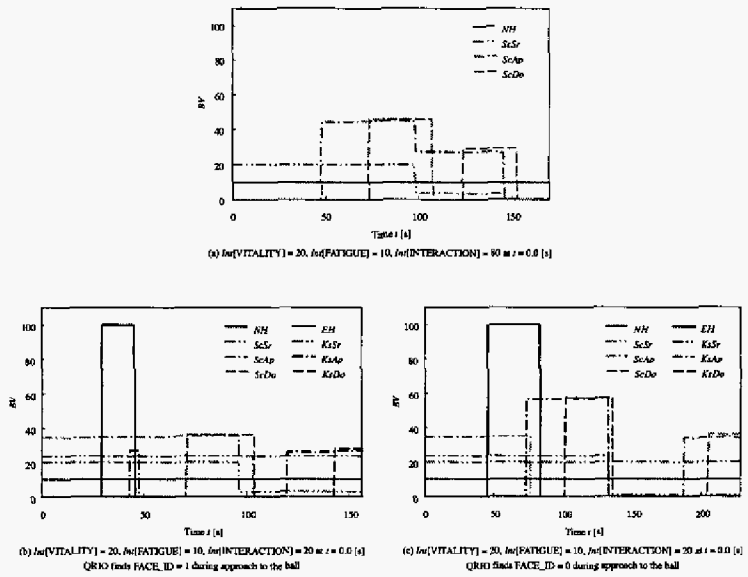


Fig. 14 Experimental result of BV

Through these experiments, it would be guessed as follows.
QRIO prefers an user who interacts with usually to playing soccer with red ball. QRIO suspends playing soccer and requests interaction for that reason. But QRIO does not prefer a user who rarely interacts with to playing soccer with red ball. QRIO ignores him and keeps playing soccer. QRIO gets attached to a user who has much interaction with QRIO. The guess would attract users to interaction with QRIO.

5. RELATED WORKS AND DISCUSSION

In the previous section, we describe learning of behavior values, which could be considered as action selection mechanism of autonomous robots. Humphrys[11] proposed learnt action selection mechanism with reinforcement learning. As we mentioned in the introduction section, the action selection mechanism is based on "releaser[2]", which are coded and debugged manually. Humphrys also addressed that action selection algorithms are mainly done by time consuming hand tuning, and little work has been done on solving the action selection problem using learning. He also pointed out that general Reinforcement Learning work has

concentrated with one evaluation function or one goal; however, there are many goals that should be considered in real world and real situations. Thus, action selection has to deal with multiple goals in a parallel execution fashion. Humphrys uses "house robot" as a use case, which has to pick up dirt and to return to some base to re-charge and empty its bag, etc. Humphrys uses Reinforcement Learning algorithm with predicted rewards and actual rewards. There are multiple rewards corresponding to actions, which are learnt by any executions of actions. Then, the simplest action selection mechanism is to select the action with the maximum reward. There are some alternatives proposed such as selecting the action that maximizing the collection of all rewards, and so on.

Our approach described in this paper can be also considered as Reinforcement Learning, but we use regulation mechanism of the internal variables as reward system. Then, the learning rule is to learn the expectation of the change of the internal variables, by which each action can compute the expected reward value based on the regulation mechanism. The merit of this approach is that the reward values depend on both of the internal states and external states. So, even if the external situation is good for a particular reward, if it is not proper in terms of the regulation rule, the expected reward value is low.

In our EGO architecture, the monitor function computes its behavior value based on the regulation mechanism of the internal variables. It can be considered that the monitor function computes the expected reward of the corresponding behavior based on the regulation mechanism. In MOSAIC architecture[12], multiple pairs of predictors and controllers are organized. In the MOSAIC architecture, a proper controller is selected based on the performance of the corresponding predictor. Thus, the predictor can be considered as the monitor function in our EGO architecture. In our EGO architecture, the behavior modules usually perform more abstract level of behavior than the controller in the MOSAIC architecture. Regarding "motivation" of the behavior, the EGO architecture handles multiple motivations based on the regulation rule of the internal variables, however, in the MOSAIC, the prediction error can be considered as a general internal variable for motivation of the behavior.

Because each behavior module has database, expectations of change in FATIGUE for approach ball and user might be different even if they are learned. They should be linked from the view point of approach behavior. Therefore implementation of database for relationship among target object, expectation of change in the internal state variable and behavior should be considered. And expectation of change in the internal state variable is learned from only a target object as external stimuli. The learning from multi dimensional external stimuli is one of our future works.

6. SUMMARY

In this paper, we describe the learning algorithm of the behavior values for behavior selection problem. The essential of the learning to make associations of

the triples (Behavior, Target, Change of Internal Variables), so that each behavior module can predict the internal variables after the behavior is executed. Then, based on the regulation mechanism of the internal variables each behavior can compute the behavior value in a situation.
We implement this algorithm using QRIO, and confirm that the learning results in different behavior tendency. For a friendly user the interaction behavior is often selected, but for an unfriendly user other behaviors are selected, and so on.

Acknowledgements

We greatly appreciate Dr. Ronald C. Arkin in Georgia Institute of Technology for his discussion of the architecture, and all researchers and engineers for QRIO in Sony Corporation for their kind corporations.

References

Proceedings:

1. M. Fujita, Y. Kuroki, T. Ishida and T. Doi, A small humanoid robot SDR-4X for entertainment applications, *Int. Conf. on Advanced Intelligent Mechatronics (AIM)* (Kobe, JPN, 2003), pp.938-943

Authored book:

2. Robin R. Murphy, Introduction to AI ROBOTICS, The MIT Press, 2000

Proceedings:

3. M. Fujita, R. Hasegawa, C. Gabriel, T. Takagi, J. Yokono and H. Shimomura, An Autonomous Robot that eats information via interaction with human and environment, *Int. Workshop on Robot-Human Interactive Communication (ROMAN)* (Bordeaux and Paris, FRN, 2001), pp.383-389.

Proceedings:

4. Fujita M., et. al., Physically and Emotionally grounded symbol acquisition for autonomous robots, *AAAI Fall Symposium: Emotional and Intelligent II* (Massachusetts, USA, 2001), pp.43-46

Proceedings:

5. R. Arkin, M. Fujita, T. Takagi and R. Hasegawa, Ethological Modeling and Architecture for an Entertainment Robot, *IEEE/RSJ Int. Conf. on Robotics and Automation (ICRA)* (Seoul, KOR, 2001)

Proceedings:

6. M. Fujita, Y. Kuroki, T. Ishida and T. Doi, Autonomous behavior control architecture

of entertainment humanoid robot SDR-4X, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (IEEE Press, LA, USA, 2003), pp.960-967

Authored book:

7. Ekman, P. and Davidson, R. J., The nature of emotion, Oxford University Press, 1994

Proceedings:

8. F. Tanaka, K. Noda, T. Sawada and M. Fujita, Associated Emotion and its Expression in an Entertainment Robot QRIO, *IFIP Int. Conf. Entertainment Computing (ICEC)* (Eindhoven, The Netherlands, 2004), in press.

Proceedings:

9. Y. Hoshino, T. Takagi and M. Fujita, Behavior description and control using behavior module for personal robot, *IEEE/RSJ Int. Conf. on Robotics and Automation (ICRA)* (IEEE Press, Louisiana, USA, 2004), pp.4165-4171

Proceedings:

10. T. Sawada, T. Takagi and M. Fujita, Behavior Selection and Motion Modulation in Emotionally Grounded Architecture for QRIO SDR-4X, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (IEEE Press, Sendai, JPN, 2004), in press

Proceedings:

11. Mark Humphrys, Action Selection methods using Reinforcement Learning, *Int. Conf. of Simulation of Adaptive Behavior (SAB)* (1996), pp.135-144

Proceedings:

12. K. Doya, K. Samejima, K. Katagiria and M. Kawato, Multiple model-based reinforcement learning,.Neural Computation, 2002, pp.1347-1369

Tsutomu Sawada received his M.S. and Ph.D. degrees from Science University of Tokyo, Japan, in 1998 and 2001, respectively. From 2001 he was researcher at the Digital Creatures Laboratory, Sony Corporation, and is currently working on development of behavior control architecture for entertainment robot SDR-4XII QRIO at Intelligent Systems Research Laboratory, Information Technologies Laboratories, Sony Corporation.

His research interests include reinforcement learning, design of sensory motor coordination and morphology, behavior control architecture and inferential system for human-machine interaction.

Tsuyoshi Takagi received a B.A in Mechanism from the Keio University, Tokyo, in 1994 and M.S. degree in mechanical engineering from the Keio University, Tokyo, in 1996. He joined Robot Entertainment project from 1998, and developed entertainment robot AIBO, which was started to sell in 1999. After the AIBO project, he worked for development of behavior control architecture based on ethology at the Digital Creatures Laboratory, Sony Corporation, and is currently working for development of behavior control architecture for entertainment robot at Entertainment Robot Company, Sony Corporation. His research interests include agent architecture, autonomous behavior, evolutionary systems, cognitive science, social interaction, and learning.

Yukiko Hoshino received her M.S. and her Ph.D. degree in Mechano-Informatics from the University of Tokyo, Japan, in 1998 and 2001, respectively. From 2001, she was at the Digital Creatures Laboratory, Sony Corporation, and is currently at the Life Dynamics Laboratory Preparatory Office, Sony Corporation.

Yukiko Hoshino works in the human robot interaction and robot behavior system, and recent work is the development of behavior selection architecture and human robot interaction for QRIO SDR-4XII. Her research interests include human-robot interaction, embodiment and behavior coordination of the robot. Also, she received the 13th Best Paper Award from The Robotics Society of Japan, in 1999, for the work of full-body tactile sensor suit, and also received the 11th Young Investigator Excellence Award from the Robotics Society of Japan, in 1996.

Masahiro Fujita is a General Manager/Chief Researcher at Information Technologies Laboratories and a Research Director at Life Dynamics Laboratory Preparatory Office in Sony Corporation. He received a B.A. degree in Electronics and Communications from the Waseda University, Tokyo, in1981, and joined Sony Corporation. He worked for development of a spread spectrum communication system, which was used for global positioning system in car navigation. From 1988, he became a graduate student of University of California, Irvine, and studied artificial neural network for visual perception. He received an M.S. degree in Electrical Engineering from the University of California, Irvine, in 1989. He started Robot Entertainment project from 1993, and developed entertainment robot AIBO, which was started to sell in 1999. After the AIBO project, he has been in charge of development for cognitive part of a small humanoid robot QRIO. His research interests include computer vision, verbal and non-verbal interaction, language acquisition, emotional model of autonomous agents, and behavior control architecture.