

Paper:

Decision Making for a Mobile Robot Using Potential Function

Kazumi Oikawa*, Hidenori Takauji**, Takanori Emaru***,
Takeshi Tsuchiya****, and Shigenori Okubo*

*Yamagata Univ., 4-3-16 Jonan, Yonezawa, Yamagata 992-8510, Japan

E-mail: okazu@yz.yamagata-u.ac.jp

**Hokkaido Univ., Kita 14 Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan

***Hokkaido Univ., Kita 13 Nishi 9, Kita-ku, Sapporo, Hokkaido 060-8628, Japan

****Hokkaido Institute of Technology, 7-15-4-1 Maeda, Teine-ku, Sapporo, Hokkaido 006-8585, Japan

[Received October 20, 2006; accepted February 26, 2007]

We discuss decision making for a behavior-based robot with modules which determining robot action. The subsumption architecture (SA) arranges modules in layers, giving upper-layer module action priority over lower-layer modules. Although implementation is easy, results in many inefficient actions because upper-layer module are used regardless of other modules. We solve this problem by representing actions by Potential Function (PF), in which maximum votes are collected from modules. Using event-driven state transition, the robot decides its action with appropriate sets of modules changed based on the situation. We apply this to navigation tasks in a corridor and show simulation results. When we give a map and path designation to the robot, we use a handwriting map interface. We compare object-oriented design SA and PMF with our proposal and show how inefficient actions are reduced using our proposal.

Keywords: potential function, autonomous mobile robot, indoor navigation, behavior-based robotics, graph map

1. Introduction

We discuss how a robot with a network of multiple modules adjusts competing modules for different decisions and develops behavior. Brooks [1] proposed the subsumption architecture (SA) using a layered network consisting of upper-layer modules having priorities among competing modules. Many behavior-based methods [3, 4] select behavior from if-then routines based on sensor information as a conditional clause. We propose an object-oriented design [8] using this framework, and applied it to a navigation task [6]. We classify SA in object-oriented design into two types, i.e., interference and noninterference, and take an approach that changes layered control based on conditions to solved problems in which noninterference SA is not suitable for tasks requiring planning, while programming is easier than in interference SA. This does not, however, basically solve the problem of developing inefficient behavior. Noninterfer-

ence SA is simply structured but tends to develop inefficient behavior because upper-layer modules are automatically prioritized in intermodule competition without overall efficiency being considered. This is due to fixed layered control implying that changes of layered control will not be able to solve the problem as long as noninterference SA is involved.

In contrast to behavior-based approaches, Nakamura et al. [5] proposed integrative reactive behavior (modules) using a smooth nonlinear function of sensor signals instead of selecting behavior discretely and selectively using if-then rules, and demonstrated object grasping using a multi-fingered hand. The intensity of reactive behavior is obtained by learning to improve robustness against environmental changes. Other proposed approaches include selecting appropriate behavior by learning, using a neural network [11], and using reinforcement learning [10], but Nakamura's approach is integrative, rather than selective, making it different from others. Calculation for expression intensity requires smooth nonlinear functions of sensor signals, and it is difficult for our system, using a microcomputer poor in processing power, to make real-time floating-point calculation using mathematical functions for all sensor signals. We found it effective to integrate modules rather than to select them in intermodule competition. When parameters are given by users rather than by learning, robustness against unexpected environmental change is decreased, so we focused on fuzzy logic giving moderate performance but enabling vague design. Although Nakamura's approach does not explicitly touch on fuzzy logic, it handles continuous logic values of [0, 1].

Using fuzzy logic, Tsuzaki et al. [13] expressed multiple rules with a potential membership function (PMF), and proposed fuzzy potential by integrating them, applying this to a RoboCup soccer robot demonstrating robustness in dynamic and complex environments. Otsuka et al. [9] realized running-round behavior corresponding to the speed of the ball by changing the PMF to match the environment. PMF, which determines the direction and speed of movement, is applied easily to our robot.

We propose determining behavior expression by composing potential functions from individual modules for intermodule competition. For activation and restraint func-

tions, a problem arises in that logical OR results for activation functions differs from intuition. We studied composing potentials apart from fuzzy logic. PMF does not handle logical OR because only one activation function is involved and others are all restraint functions. Nakamura's study includes logical OR and AND of activation and restraint functions, but does not discuss either the meaning of composition results nor the problem in composition using logical OR. We first describe the robot, then we discuss problems in selective behavior decisions by clarifying why noninterference SA selects inappropriate behavior. We then discuss problems with fuzzy composition and propose the potential function (PF) for composition without using fuzzy logic. We demonstrate PF effectiveness over other approaches in a navigation task for an autonomous mobile robot.

2. System Configuration and Working Environment

In study a navigation task problem, we use graph expression as an internal map held by the robot for solving the problem. Due to a lower amount of information compared to absolute coordinate expression, graph expression may be inferior in detailing but this factor reduces calculation because it does not require detailed path planning. Graph maps are effective in man-robot interfacing. When we navigate in daily life, we are not conscious of detailed coordinates. This is also true when indicating a destination to another person. Features (local landmarks) in environments are mostly used to indicate a destination. Although there may be some exception, those being given instructions also draw graphic maps in their minds connecting features for planning. Detailed planning and behavior decisions are conducted on an arc when a plan is executed. In short, the graph map is easy for people to use. For these reasons, we use graph map expression for a corridor environment with local landmarks. The local landmark, or simply landmark, is defined as being observable only from neighboring areas.

2.1. Mobile Robot

We conducted simulation assuming an independent wheel-driven mobile robot with 8 position-sensitive detectors (PSDs) as distance sensors, 1 electronic compass, and a landmark sensor for receiving landmark signals from 7 directions, including directly above itself. The CPU is an H8/3064. The robot is assumed not to use dead reckoning and cannot locate itself precisely (**Fig. 1**).

2.2. Working Environment

As in previous studies [6,8], cross-road and T-intersections and landmarks at endpoints are distributed in a corridor (**Fig. 2**). Although landmarks are preferably any arbitrary features in the environment, we used special artificial objects easy for robots to detect, considering the

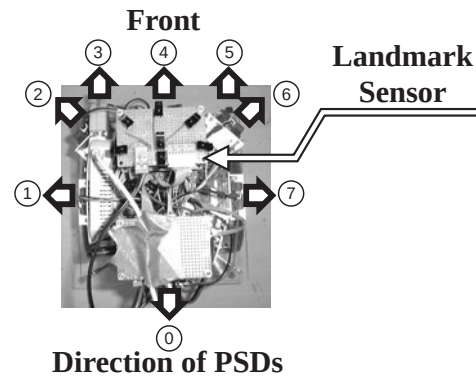


Fig. 1. Mobile robot.



Fig. 2. Working environment.

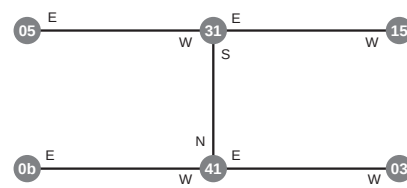


Fig. 3. Graph map.

technical issues of sensors and microcomputers incorporated. We assume that landmarks emit infrared ID signals and that the robot recognizes IDs within the reach of signals. ID is expressed in hexadecimal in figures and below.

2.3. Graph Map

The graph map for **Fig. 2** is shown in **Fig. 3**. Arcs in the graph indicate corridors and nodes indicate landmarks. The node degree is a maximum of four and restriction arise from the sensor issue on the robot. To lead the robot toward intended corridors without choosing incorrect way at cross-road and T-intersections, directional information is given in arcs. The direction may, however, differ from the actual direction. When a robot uses this map, it must be able to make its actual directions correspond to directions on the map.

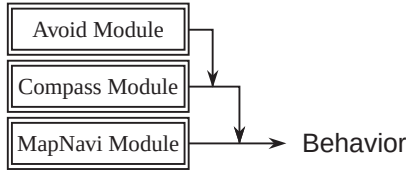


Fig. 4. Layered control.

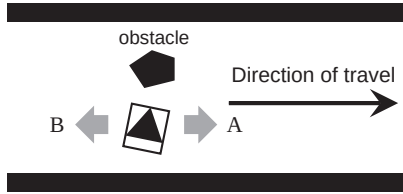


Fig. 5. Example of problem of SA.

3. Problems in Noninterference SA

Noninterference SA is a behavior decision in layered control (Fig. 4), and modules outputting behavior are free from interference from other modules. For this reason, it does not require complex processing because upper-layer modules have priority even when module behavior is mutually contradictory. With this simplified control, however, optimized behavior at the module level often leads to inappropriate behavior overall. When the robot (Fig. 4) is placed in a circumstance (Fig. 5) such as “Avoid Module” for avoiding obstacles, it can only select avoidance behavior in direction A or B. “Compass Module” for traversing the corridor, however, selects direction A. “Avoid Module,” isolated from information from other modules, must decide A or B on its own. Assuming that it selects direction B, the decision conflicts with “Compass Module,” but decision B from an upper-layer module, “Avoid Module,” is prioritized and selected.

As stated above, layered control in which the upper layer is prioritized tends to produce inappropriate behavior due to the dogmatic decision by the upper layer ignoring intentions of lower layers. Metaphorically speaking, noninterference SA is feudalistic behavior, as opposed to integrative methods including PMF that are democratic. Studies [6, 8] solve this problem by changing layered control based on the individual situation. Corridor behavior, however, is often ignored by obstacle-avoidance behavior, leaving the problem of inappropriate behavior unsolved.

4. PMF Logical OR

A problem arises when PMF logical OR results differ from intuition. We define PMF below. The lateral axis of PMF is the direction of robot movement, and the membership grade represents direction priority. When $\mu(\theta)$ means priority of θ as is, PMF is defined as activation, and when it means restraint ratio, PMF is defined as restrain-

ing, expressed by symbol $\tilde{\mu}(\theta)$. The restraint ratio varies in its meaning by arithmetic operation, but generally involves reducing priority by operation. Specifically, the restraint ratio of 1 does not reduce priority, and 0 causes priority 0, so activation and restraint PMFs are meaningful only in the following three operations:

$$\begin{aligned}\mu(\theta) &= \mu_1(\theta) \vee \mu_2(\theta) \\ \mu(\theta) &= \mu_1(\theta) \wedge \mu_2(\theta) \quad (1) \\ \bar{\mu}(\theta) &= \bar{\mu}_1(\theta) \wedge \bar{\mu}_2(\theta)\end{aligned}$$

where, in composing potentials, $\vee$, $\wedge$ can be s-norm, t-norm defined in fuzzy logic other than logical OR, AND.

Consider the case of logical OR of two activation PMF μ_1 and μ_2 for θ_1 and θ_2 . When $\mu_1(\theta_1) = \mu_2(\theta_1) = \mu_1(\theta_2) = 1.0$, $\mu_2(\theta_2) = 0.0$, logical OR is as follows:

$$\begin{aligned}\mu_1(\theta_1) \vee \mu_2(\theta_1) &= 1.0 \\ \mu_1(\theta_2) \vee \mu_2(\theta_2) &= 1.0.\end{aligned}$$

The same results are obtained in algebraic sum, limiting sum, and drastic sum. Because $\mu_2(\theta_2) = 0.0$, it is intuitively natural that priority may higher with θ_2 than θ_1 . The discrepancy between the operation result and intuition is because fuzzy logic uses a continuous logical value of $[0, 1]$, i.e., the problem arises because results of logical OR saturate at 1.

5. Potential Function Approach

Taking the above example into account, a simple sum operation matches our intuition for operation between activation PMFs. The result of operation may exceed the range $[0, 1]$, so we do not handle fuzzy logic here. We propose PF in which where potential functions are not expressed with membership functions.

5.1. Potential Function

We discuss composing operations from activation and restraint function output from individual modules. If we handle activation and restraint functions separately, it could produce a design that may produce contradictory output such as priority 1 and restraint ratio 0 for a certain θ , so we integrated the two functions.

As in PMF, the lateral axis indicates relative angle $\Theta = \{\theta_1, \dots, \theta_n\}$ in PF. The vertical axis represents priority and restraint ratio as follows:

$$f : \Theta \mapsto [-1, 1]; \quad f(\theta_i) \in [-1, 1], \theta_i \in \Theta. \quad . . . (2)$$

When $f(\theta_i) > 0$, f becomes active. When $f(\theta_i) \leq 0$, it becomes restraining, and the resulting value subtracting the absolute value of it from 1 is equal to the restraint ratio.

5.2. Composing PF

In addition to PF, activation function F and restraint function $\bar{F}$ for composing operation are defined as fol-

lows:

$$F: \Theta \mapsto [0, \infty); \quad F(\theta_i) \in [0, \infty), \theta_i \in \Theta \quad \dots \quad (3)$$

$$\bar{F}: \Theta \mapsto [0, 1]; \quad \bar{F}(\theta_i) \in [0, 1], \theta_i \in \Theta. \quad \dots \quad (4)$$

We define the following functions:

$$\delta_+(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad \dots \quad (5)$$

$$\delta_-(x) = \begin{cases} 0 & (x \geq 0) \\ -x & (x < 0) \end{cases} \quad \dots \quad (6)$$

The composing operation with m sets of PFs is defined as follows:

$$F(\theta_i) = \sum_{j=1}^m \delta_+(f_j(\theta_i)) \quad \dots \quad (7)$$

$$\bar{F}(\theta_i) = \prod_{j=1}^m \{1 - \delta_-(f_j(\theta_i))\} \quad \dots \quad (8)$$

$$F^*(\theta_i) = F(\theta_i) \times \bar{F}(\theta_i) \quad \dots \quad (9)$$

where F^* represents a composed PF with the value of $[0, m]$, and $F^*(\theta_i)$ represents the final priority of θ_i .

Direction of movement θ of the robot is obtained using appropriate functions. Here we use defuzzification BADD used elsewhere [13] and expressed as follows:

$$\theta = \frac{\sum_{i=1}^n (F^*(\theta_i))^\lambda \theta_i}{\sum_{i=1}^n (F^*(\theta_i))^\lambda} \quad \dots \quad (10)$$

Defuzzification adjusts the degree of evaluation for the maximum value by λ , in which the greater the λ values, the greater the weight of the maximum value. While the mean of maxima evaluates only the maximum value, BADD evaluates other candidates around the maximum value, which is what we used BADD for. We used $\lambda = 20$ based on the study [13].

5.3. Application to Mobile Robot

Parameters required for an independent wheel-driven mobile robot to operate are forward speed v and turning speed ω . To decide these independently, direction θ and distance ϕ in open space are used [2]. θ is obtained from the result of composing PF, then ϕ is obtained from θ . Transform function (TF) from θ to ϕ is defined as follows:

$$G: \Theta \mapsto [0, \infty); \quad G(\theta_i) \in [0, \infty), \theta_i \in \Theta. \quad \dots \quad (11)$$

TF is obtained from a composing operation as in PF. TF output from modules is defined in the same way. To distinguish between them, the composed TF is expressed in uppercase and the TF from modules in lowercase. A composing operation with m sets of TF is defined as follows:

$$G^*(\theta_i) = \min(g_1(\theta_i), g_2(\theta_i), \dots, g_m(\theta_i)). \quad \dots \quad (12)$$

Distance ϕ is obtained from the composed TF as follows:

$$\phi = G^*(\theta). \quad \dots \quad (13)$$

Forward speed v and turning speed ω are then obtained as follows:

$$v = K_v \phi \quad \dots \quad (14)$$

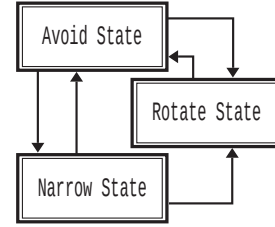


Fig. 6. State transition diagram of Avoid Module.

$$\omega = K_\omega \theta \quad \dots \quad (15)$$

where K_v and K_ω represent gain constants.

5.4. Module Design

Modules designed for executing navigation tasks are described here. As stated earlier, we assume the use of a robot we built in our laboratory, and prepared an Avoid Module for avoiding obstacles, a Compass Module for traversing the corridor, and a MapNavi Module for navigation, as outlined below for PF and TP.

5.4.1. Avoid Module

Obtains directions for obstacles and open space from information from distance sensors, generating PF to avoid obstacles (Fig. 6).

Avoid State: The state in which basic obstacle-avoidance behavior is generated. When detecting a wall ahead, the robot transits to the Rotate State. When detecting a wall on both sides, it transits to Narrow State.

Rotate State: The state in which the robot rotates until it faces an open space, at which point it transits to Avoid State.

Narrow State: The state in which behavior is generated for passing through a narrow space. When detecting a wall ahead, the robot transits to Rotate State. When no longer detecting a wall on either sides, it transits to Avoid State.

In generating PF, potential p is transformed from distance x [cm] to an obstacle using the following equation:

$$p = \begin{cases} -1 & (x < 20) \\ \frac{(x-25)}{5} & (20 \leq x < 25) \\ \frac{(x-25)}{15} & (25 \leq x < 40) \\ 1 & (40 \leq x) \end{cases} \quad \dots \quad (16)$$

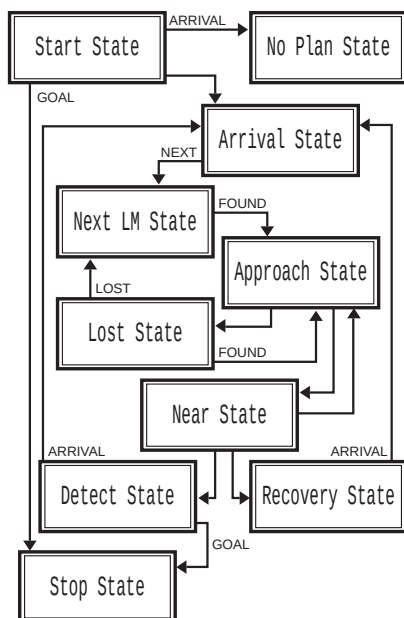
PF is processed based on the environment. To avoid conflict with obstacles during turning, TF is transformed with a weight using Table 1. This is done in other than Rotate State, in which TF becomes 0 in all directions to makes rotation.

5.4.2. Compass Module

The robot incorporates far fewer sensors to traverse the corridor, unable to detect longitudinal directions. This

Table 1. Weighting function.

$\theta[\text{rad}]$	$-\pi$	$-3\pi/4$	$-\pi/2$	$-\pi/4$
$W(\theta)$	0.00	0.25	0.50	0.75
$\theta[\text{rad}]$	0	$\pi/4$	$\pi/2$	$3\pi/4$
$W(\theta)$	1.00	0.75	0.50	0.25

**Fig. 7.** State transition diagram of MapNavi Module.

module uses the earth's magnetism to lead the robot to the destination, enabling to traverse the corridor. The desired direction is set to the compass value when the module is initialized. The module does not conduct state transition. PF simply gives the maximum value to the desired direction. TF gives the maximum value to the front and this is fixed.

5.4.3. MapNavi Module

Leads the robot to destinations based on the instructed path using a graph map given by the manuscript map interface (**Fig. 7**).

Start State: The state in which initial settings are done. If a path instruction is given, the robot transits to Arrival State. If not, it transits to No Plan State generating event ARRIVAL. At this moment, if the robot is at the destination, it transits to Stop State generating event GOAL.

Arrival State: The state in which the robot has reached a landmark. After rotating to the next landmark, the robot transits to Next LM State generating event NEXT.

Next LM State: The state in which the robot is on the way to the next landmark. When entering a landmark-observable area, the robot transits to Approach State generating event FOUND.

Approach State: The state in which the robot moves toward a near zone in the observable area. When detecting a near zone of a landmark, the robot transits to Near State. When losing the landmark, it transits to Lost State.

Near State: The state in which the robot is approaching the landmark identifying ID detected by the sensor for near zones. When detecting the next landmark ID, the robot transits to Detect State. When other IDs are detected, it transits to Recovery State.

Lost State: The state in which the robot has lost the landmark while approaching the landmark. If unable to detect in a certain period, the robot transits to Next LM State generating event LOST. If detected, it transits to Approach State.

Detect State: The state in which the robot identified a landmark ID and is making a subplan for the next landmark. If the robot has reached the final destination, it transits to Stop State generating event GOAL. If not, it makes a subplan and transits to Arrival State generating event ARRIVAL.

Recovery State: The state in which the robot is recovering from an error such that it reached an unintended landmark. After making a subplan to the intended landmark, the robot transits to Arrival State generating event ARRIVAL.

Stop State: The state in which the robot has reached the destination and stops.

No Plan State: The state in which no path instruction is given.

PF is generated to become 0 in all directions when the robot is outside landmark-observable areas, and becomes maximum in the direction of a landmark when inside the zones. TF is generated in the same way as in Compass Module.

5.5. Example of PF and TF Operations

With the modules above, we describe how PF and TF are output and composed in examples. **Fig. 8** shows a simulation scene. The square with a triangle mark means the robot and its posture. The direction of the triangle mark indicates the front of the robot. A cluster in black indicates a wall or obstacle, and the remainder corridors. A number in a corridor indicates a landmark ID and the location. Area in gray around the landmark means the landmark-observable area. The white arrow indicates the desired direction for the robot. With these conditions, PF output from each module and activation and restraint functions F , $\bar{F}$ obtained from PF are shown in **Table 2**. TF and composed results are shown in **Table 3**. Finally-obtained composed function F^* is shown in **Table 4**. For comparison to PMF, composed functions obtained from activation functions using fuzzy logical OR ($\vee$), algebraic

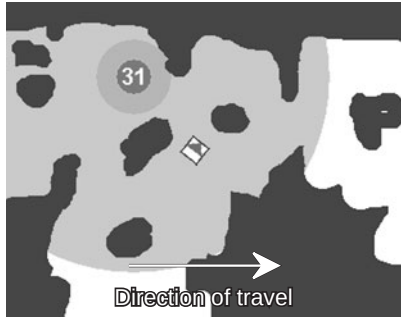


Fig. 8. A scene of navigation.

Table 2. PFs of each module.

$\theta[\text{rad}]$	$-\pi$	$-3\pi/4$	$-\pi/2$	$-\pi/4$
f_{Avoid}	0.00	-1.00	-1.00	0.80
f_{Compass}	0.00	0.00	0.00	0.20
f_{MapNavi}	0.00	0.20	0.50	1.00
F	0.00	0.20	0.50	2.00
F	1.00	0.00	0.00	1.00

$\theta[\text{rad}]$	0	$\pi/4$	$\pi/2$	$3\pi/4$
f_{Avoid}	-1.00	0.60	0.50	0.00
f_{Compass}	0.50	1.00	0.50	0.20
f_{MapNavi}	0.50	0.20	0.00	0.00
F	1.00	1.80	1.00	0.20
F	0.00	1.00	1.00	1.00

Table 3. TFs of each module.

$\theta[\text{rad}]$	$-\pi$	$-3\pi/4$	$-\pi/2$	$-\pi/4$
g_{Avoid}	0	0	9	40
g_{Compass}	0	0	5	75
g_{MapNavi}	0	0	3	53
G^*	0	0	3	40

$\theta[\text{rad}]$	0	$\pi/4$	$\pi/2$	$3\pi/4$
g_{Avoid}	20	64	25	0
g_{Compass}	100	75	5	0
g_{MapNavi}	70	53	3	0
G^*	20	53	3	0

sum ($\dot{+}$), and limiting sum ($\oplus$). Consequently, θ [rad] obtained from each composed function using Eq. (10) is as follows:

$$\begin{aligned}\theta_{F^*} &\approx -0.615 \quad (\text{PF}) \\ \theta_{\vee} &\approx 0.000 \quad (\text{logical OR}) \\ \theta_{\dot{+}} &\approx 0.002 \quad (\text{algebraic sum}) \\ \theta_{\oplus} &\approx 0.524 \quad (\text{limiting sum}).\end{aligned}$$

As results indicate, PF selects a left turn evaluating large potentials from Avoid Module and MapNavi Module, while a method using fuzzy logic causes a straight forwarding because the large potentials on the left and right are saturated and evaluated in the same degree causing the middle way to be selected. Limiting sum selected

Table 4. Composition results.

$\theta[\text{rad}]$	$-\pi$	$-3\pi/4$	$-\pi/2$	$-\pi/4$
F^*	0.00	0.00	0.00	2.00
$\vee$	0.00	0.00	0.00	1.00
$\dot{+}$	0.00	0.00	0.00	1.00
$\oplus$	0.00	0.00	0.00	1.00

$\theta[\text{rad}]$	0	$\pi/4$	$\pi/2$	$3\pi/4$
F^*	0.00	1.80	1.00	0.20
$\vee$	0.00	1.00	0.50	0.20
$\dot{+}$	0.00	1.00	0.75	0.20
$\oplus$	0.00	1.00	1.00	0.20

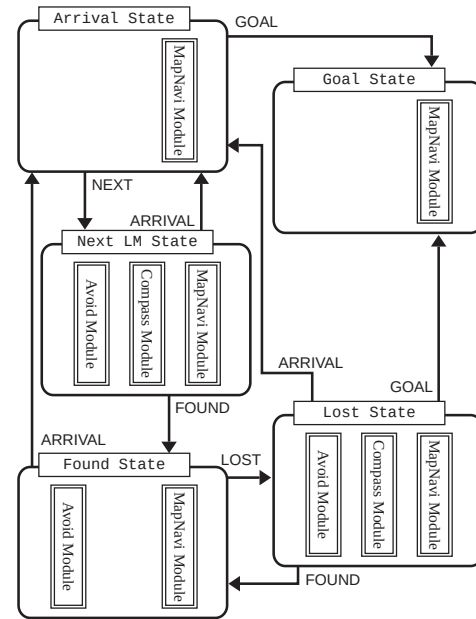


Fig. 9. Event driven state transition diagram with PF.

even a right turn. As apparent from Fig. 8, these selections are not the desired decision. These results demonstrate that total sum gives better results in operations for multiple activation functions.

5.6. Event-Driven Layered Control Change

To execute tasks smoothly, modules must be selected based on the situation. For this, we use the event-driven state transition used in object-oriented design SA [8]. Navigation design is shown in Fig. 9. A word at the base of an arrow in the picture indicates the name of an event that triggers a state transition.

Arrival State: The state in which the robot has reached a landmark. The state consists of MapNavi Module alone to eliminate the influence of other modules, enabling the robot to rotate to the direction of the next landmark.

Next LM State: The state in which the robot is on the way to the next landmark. When entering this state,

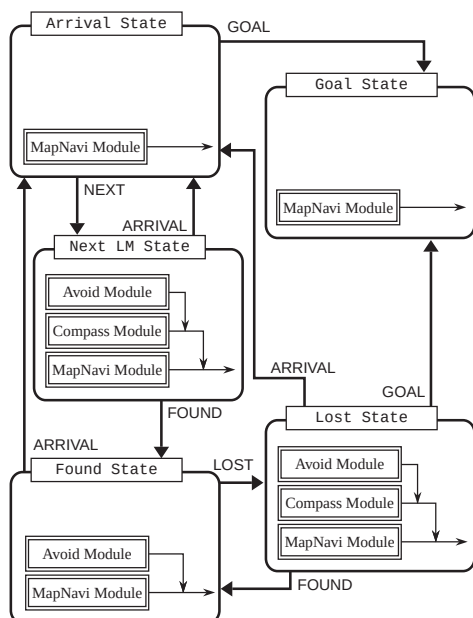


Fig. 10. Event driven state transition diagram with SA.

it initializes Compass Module setting the direction to the desired direction.

Found State: The state in which the robot enters a landmark-observable area. Because no corridor need be traversed, Compass Module is not incorporated.

Lost State: The state in which the robot has lost the landmark.

Goal State: The state in which the robot reached the final destination. Upon arrival at the destination, neither avoiding obstacle nor traversing a corridor is required, so only MapNavi Module is incorporated.

6. Simulation

To verify the effectiveness of PF, we conducted a comparison experiment with PMF and SA on a computer, comparing the three approaches for efficiency. The number of steps from start to goal is used as the index. The step means the sampling cycle, which is 50 ms here. The number of steps, however, is influenced by travel speed, and attention must be paid so that efficiency is not determined by the number of steps alone. The comparison here involves the judgment of whether it chooses the appropriate speed. It holds here that the fewer the steps, the greater the efficiency.

6.1. Setting in SA and PMF

SA is designed based on the object-oriented design reported in [8]. To increase comparison precision, modules designed by PF are adapted to layered control. The state

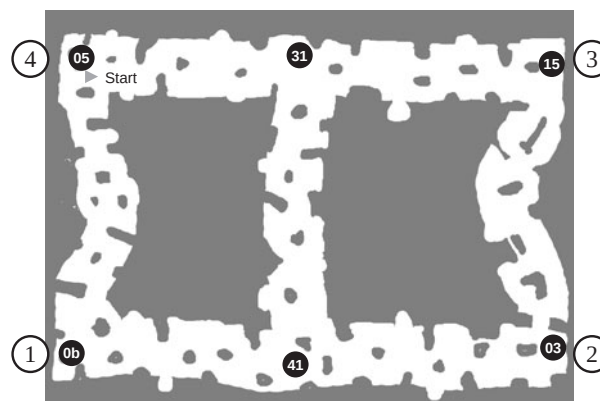


Fig. 11. Corridor environment of simulation.

Table 5. The simulation result of each approach.

	Ave.	SD	min	max
PF	1534.1	103.0	1431	1726
PMF	1758.4	96.7	1626	1985
SA	1975.3	534.5	1623	3679

transition designed here is shown in **Fig. 10**. In the operation for obtaining activation functions in PF for PMF, we used fuzzy algebraic sums to include summation. Otherwise, it is the same as in PF.

6.2. Navigation Task

Figure 11 shows the working environment. The solid black circle represents the position of a landmark, and the number (hexadecimal) in white indicates a landmark ID. The triangle near the word “Start” indicates the initial location of the robot. Numbers on both sides of the map are the sequence of landmarks for the robot to follow. The graph map for the environment was created from a handwritten linear drawing using automatic conversion [7].

6.3. Results and Discussion

Average, standard deviation, and maximum and minimum numbers of steps are shown in **Table 5** with PF, PMF, and SA for 20 trials each. Welch's t-test showed that PF results are significant against that of PMF and SA, proving that PF requires fewer steps to a destination than the other two. Worst results taking the maximum number of steps with PF, PMF and SA are shown in **Figs. 12-14**. Locations and directions of the robot every 5 steps are indicated. Numbers indicate the same as in **Fig. 11** and 0 means the initial location.

The difference between PF and PMF is observed at A and B points in **Figs. 12** and **13**. With PF, the robot goes straight to the landmark within the landmark-observable area, while it approaches the landmark following a round-about trajectory with PMF due to the fact that activation potentials saturated at 1, as observed in section 5.5, and all peaks are evaluated as the same. This caused higher

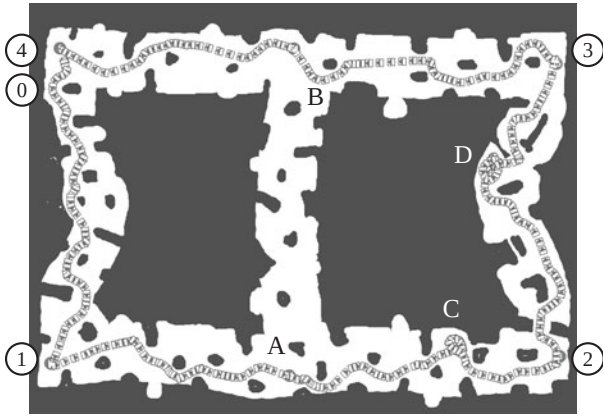


Fig. 12. Horrible result with PF.

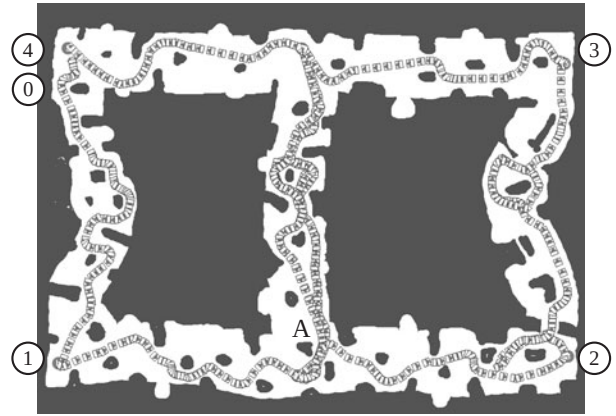


Fig. 14. Horrible result with object-oriented design SA.

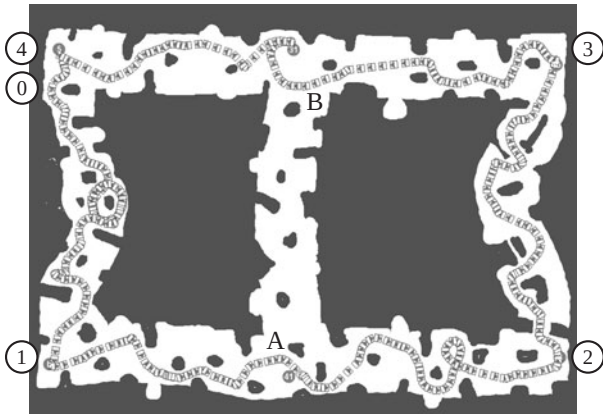


Fig. 13. Horrible result with PMF.

efficiency with PF. SA did the worst occasionally entering incorrect corridors. At point A in Fig. 14, Avoid Module was preferentially selected, ignoring output from Compass Module, resulting in entering incorrect corridors. The proposal in the study [6, 8] results in such situations, which, as stated in section 3, is caused by selective module decisions. PF did not cause this problem in this environment. This also demonstrates the superiority of the integrative approach. In short, simulation results proved that the integrative approach effectively adjusts competing modules, and that the method summing priorities will develop more appropriate behavior, which also matches our intuition, than a fuzzy logic framework, in composing operation from activation potential functions. PF also develops inefficient behavior as observed at C and D in Fig. 12, partly due to the low angular resolution of distance sensors (only 8 directions for surrounding areas), meaning using higher-resolution sensors could improve its performance. Another reason is that this approach determines behavior only with partially observable information. Based on a combination of obstacles, the robot may enter an incorrect corridor, which did not happen this time. This is a big problem when relying on graph maps with less information and partial observation. Using information-rich map expression as in study [12]

Table 6. The simulation result of PF(MOM).

	Ave.	SD	min	max
PF(MOM)	1542.3	61.3	1445	1661

may solve the problem. We consider a case in which only relations between features are given as was given in verbal instructions, and will continue our study to achieve the objective with less information or find limitations.

We used BADD defuzzification to obtain direction of movement θ , which is difficult to calculate in real time with our robot's CPU. Since composed function is $F^* \leq m$ for module number m , there may be a way to normalize F^* by m and place results calculated offline into an array. This may cause a problem if memory capacity is insufficient. As a reference, we present results of 20 trials of the above task using the mean of maxima (MOM) in Table 6. MOM, which does not require exponential operation, could be realized on our robot, and no big difference is found in results. Although evaluation for maximum values can be adjusted by λ to reflect the designer's intention in BADD defuzzification, MOM may be a choice when a less powerful CPU is used.

7. Conclusion

We have proposed integrating decisions from individual modules to solve the problem in SA that develops inefficient behavior caused by selective decisions in which decisions from upper-layer modules are prioritized. We defined activation and restraint functions, and pointed out that results from fuzzy logic operation do not match intuition in composing operation from activation functions. For a solution of the problem, we proposed PF that develops behavior by composing potential functions rather than by fuzzy logic. We demonstrated in simulation that PF is superior to PMF and object-oriented design SA.

We will apply the outcome to a mobile wheeled robot to study the possibility of realize it method on real robots. We will continue to expand application fields, includ-

ing quadrupedal walking robots, which we have already started, and further study the possibility for applications beyond navigation.

References:

- [1] R. A. Brooks, "A Robust Layered Control System For A Mobile Robot," IEEE Journal of Robotics and Automation, RA-2(1), pp. 14-23, 1986.
- [2] T. Emaru, K. Tanaka, and T. Tsuchiya, "Speed control of a sonar-based mobile robot with considering the self-localization," In IEEE International Conference on Mechatronics & Automation, pp. 125-130, 2005.
- [3] J. L. Jones and A. M. Flynn, "Mobile Robots: Inspiration to Implementation," A K Peters, Ltd, 1993.
- [4] M. J. Mataric, "Integration of Representation Into Goal-Driven Behavior-Based Robots," IEEE Transaction on Robotics and Automation, 8(3), pp. 304-312, 1992.
- [5] Y. Nakamura and T. Yamazaki, "The Integration Theory of Reactive Behavior and Its Application to Reactive Grasp by a Multi-Fingered Hand," Journal of the Robotics Society of Japan, 15(3), pp. 448-459, 1997 (in Japanese).
- [6] K. Oikawa, H. Takauji, T. Emaru, S. Okubo, and T. Tsuchiya, "Navigation Using Local Landmarks in a Corridor Environment," Journal of Robotics and Mechatronics, Vol.17, No.3, pp. 262-268, 2005.
- [7] K. Oikawa, H. Takauji, T. Emaru, S. Okubo, and T. Tsuchiya, "Navigation Instructions Using Handwriting Map Interface," in Proceedings 2006 JSME Conference on Robotics and Mechatronics, 2006 (in Japanese).
- [8] K. Oikawa, T. Tsuchiya, and S. Okubo, "Object-Oriented Design of Subsumption Architecture," Journal of the Robotics Society of Japan, 23(6), pp. 697-705, 2005 (in Japanese).
- [9] F. Otsuka, H. Fujii, and K. Yoshida, "Action Control Based on Extended PMF for an Autonomous Mobile Robot," in Proceedings of the 23rd Annual Conference of the Robotics Society of Japan (CDROM), 2005 (in Japanese).
- [10] Y. Takahashi and M. Asada, "State-Action Space Construction for Multi-Layered Learning System," Journal of the Robotics Society of Japan, 21(2), pp. 164-171, 2003 (in Japanese).
- [11] J. Tani, "Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26(3), pp. 421-436, 1996.
- [12] M. Tomono and S. Yuta, "Indoor Navigation based on an Inaccurate Map using Object Recognition," Journal of the Robotics Society of Japan, 22(1), pp. 83-92, 2004 (in Japanese).
- [13] R. Tsuzaki and K. Yoshida, "Motion Control Based on Fuzzy Potential Method for Autonomous Mobile Robot with Omnidirectional Vision," Journal of the Robotics Society of Japan, 21(6), pp. 656-662, 2003 (in Japanese).



Name:
Kazumi Oikawa

Affiliation:
Assistant Professor, Yamagata University

Address:

4-3-16 Jonan, Yonezawa, Yamagata 992-8510, Japan

Brief Biographical History:

2000- Received Ph.D. from Hokkaido University
2000- Research Associate, Yamagata University
2007- Assistant Professor, Yamagata University

Main Works:

- "Navigation Using Local Landmarks in a Corridor Environment," Journal of Robotics and Mechatronics, Vol.17, No.3, pp. 262-268, 2005.
- "Object-Oriented Design of Subsumption Architecture," Journal of the Robotics Society of Japan, Vol.23, No.6, pp. 697-705, 2005 (in Japanese).

Membership in Academic Societies:

- The Japan Society of Mechanical Engineers (JSME)
- The Robotics Society of Japan (RSJ)



Name:
Hidenori Takauji

Affiliation:
Graduate School of Information Science and Technology, Hokkaido University

Address:

Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan

Brief Biographical History:

1997- The B.E. degree in Electrical Engineering, Hokkaido University, Japan
1999- The M.E. degree in Systems and Information Engineering, Hokkaido University, Japan
2006- The Ph.D. degree in Systems and Information Engineering, Hokkaido University, Japan

Main Works:

- "Robust Tagging in Strange Circumstance," IEEJ Trans. EIS, Vol.125, No.6, 2005 (in Japanese).
- "Scalable Image Searching Method based on Orientation Code Density," JSPE, Vol.72, No.4, 2006 (in Japanese).

Membership in Academic Societies:

- The Robotics Society of Japan (RSJ)
- The Japan Society for Precision Engineering (JSPE)



Name:
Takanori Emaru

Affiliation:
Associate Professor, Hokkaido University

Address:

Kita 13 Nishi 9, Kita-ku, Sapporo, Hokkaido 060-8628, Japan

Brief Biographical History:

2002- Postdoctoral Research Fellow at Hokkai-Gakuen University

2003- Postdoctoral Research Fellow of the Japan Society for the Promotion of Science

2006- Assistant Professor at Osaka Electro-Communication University

2007- Associate Professor at Hokkaido University

Main Works:

- “Research on Estimating Smoothed Value and Differential Value by Using Sliding Mode System,” IEEE Trans. on Robotics and Automation, Vol.19, No.3, pp. 391-402, 2003.

Membership in Academic Societies:

- The Institute of Electrical and Electronics Engineers (IEEE)
- The Japan Society of Mechanical Engineers (JSME)
- The Institute of Electronics, Information and Communication Engineers (IEICE)



Name:
Shigenori Okubo

Affiliation:
Professor, Yamagata University

Address:

4-3-16 Jonan, Yonezawa, Yamagata 992-8510, Japan

Brief Biographical History:

1980- Lecturer, University of Tokyo

1987- Associate Professor, University of Tokyo

1988- Associate Professor, Yamagata University

1990- Professor, Yamagata University

Main Works:

- “Design of Nonlinear Regulators Using Genetic Algorithms,” Transactions of the Society of Instrument and Control Engineers, Vol.33, No.11, pp. 1072-1080, 1997 (in Japanese).

Membership in Academic Societies:

- The Society of Instrument and Control Engineers (SICE)
- The Institute of Electrical Engineers of Japan (IEEJ)
- The Japan Society of Mechanical Engineers (JSME)



Name:
Takeshi Tsuchiya

Affiliation:
Professor, Hokkaido Institute of Technology

Address:

7-15-4-1 Maeda, Teine-ku, Sapporo, Hokkaido 006-8585, Japan

Brief Biographical History:

1966- Lecturer, Hokkaido University

1967- Associate Professor, Hokkaido University

1987- Professor, Hokkaido University

2004- Professor, Hokkaido Institute of Technology

Main Works:

- “Modern Control Engineering,” Sangyo-Tosho Pub., 1991 (in Japanese).
- “Digital Preview & Predictive Control,” Sangyo-Tosho Pub., 1992 (in Japanese).
- “Basic System Control Engineering,” Morikita-Pub., 2001 (in Japanese).
- “Mechatronics 2nd Ed.,” Morikita-Pub., 2004 (in Japanese).

Membership in Academic Societies:

- The Institute of Electrical and Electronics Engineers (IEEE)
- The Japan Society of Mechanical Engineers (JSME)
- The Institute of Electrical Engineers of Japan (IEEJ)
- The Robotics Society of Japan (RSJ)
- The Society of Instrument and Control Engineers (SICE)