



Fusion of Hierarchical Behavior-based Actions in Mobile Robot Using Fuzzy Logic

Gan Zhen Ye, and Dae-Ki Kang*, *Member, KIICE*

Department of Ubiquitous IT, Dongseo University, Busan 617-716, Korea

Abstract

This paper presents mobile robot control architecture of hierarchical behaviors, inspired by biological life. The system is reactive, highly parallel, and does not rely on representation of the environment. The behaviors of the system are designed hierarchically from the bottom-up with priority given to primitive behaviors to ensure the survivability of the robot and provide robustness to failures in higher-level behaviors. Fuzzy logic is used to perform command fusion on each behavior's output. Simulations of the proposed methodology are shown and discussed. The simulation results indicate that complex tasks can be performed by a combination of a few simple behaviors and a set of fuzzy inference rules.

Index Terms: Behavior-based system, Command fusion, Fuzzy logic, Mobile robot control architecture

I. INTRODUCTION

Behavior-based robotics is a branch of robotics control architecture. In 1986, Brooks [1] introduced subsumption architecture, where the control system is decomposed into layers. Each layer in the control system represents a different behavior and each behavior can then be further decomposed into sets of simple instructions for the actuator of the mobile robot. Behavior-based robotics can be considered to have emerged from what Brooks [1] had introduced. Since then, many autonomous robotic systems [2-6] have been described as having "behavior-based" architecture. However, Werger [7] pointed out that many systems that have been claimed to be behavior-based have deviated from Brooks [1]'s strict behavior-based approach as they often end up with systems that are highly deliberative and complicated. Based on previous studies done by [8-13], we identify a behavior-based system as a system that meets the following characteristics: highly reactive, relying

minimally on real-world representation, and highly parallel.

The result of the system should achieve a real-time response to any event. A good example that demonstrates this trait in biological life is the cockroach; it can respond to the slightest sense of danger on impulse, without needing much time to plan an escape route.

Each behavior in a behavior-based system might receive the same sensory input from the sensors and will compete for the control of the robot's actuators, as individual behaviors might produce different decisions to be sent to the actuators. Hence, there is a need to decide what action the robot should perform out of all possible actions suggested by each of the behaviors. The final action selection can be accomplished through the behavior arbitration method [4], hierarchical behavior control [5], or fusion of actions [6]. The arbitration method is the act of selecting one behavior's decision as the final decision of the robot. This method might work perfectly if there are enough behaviors in the system to represent each and every situation the mobile

Received 08 March 2012, Revised 14 April 2012, Accepted 20 April 2012

*Corresponding Author E-mail: dkkang@dongseo.ac.kr

Open Access <http://dx.doi.org/10.6109/jicce.2012.10.2.149>

print ISSN:2234-8255 online ISSN:2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

robot might encounter. However, if any unpredicted situation occurs, then the robot might end up selecting a wrong action to perform. There might be a case where the situation can be represented by more than one behavior, and by selecting only one behavior's decision, it might not be sufficient to address or solve the problem. In this study, we designed the behavior system hierarchically and selected the most suitable action using the fusion method. This way we can make use of all of the behaviors' decisions and produce the best decision for each situation encountered. This will be further discussed in section II. The simulation result is shown and discussed in section III. The conclusion of our work is presented in section IV.

II. SYSTEM DESCRIPTION

A. Mobile Robot

We designed the system based on the mobile robot shown in Fig. 1 below. The mobile robot shall have six range sensors, with a pair facing outward from each of the left, front, and right directions of the mobile robot. The sensors are grouped into three categories: left sensor, front sensor, and right sensor, with each group having two sensors. The output of each group shall be the reading of the sensor that detects an obstacle in the closest proximity to the mobile robot. If left sensor 1 detects an obstacle at x distance and left sensor 2 detects an obstacle at y distance, if x is less than y , then x shall be the output of the left sensor group. The same theory applies for the other sensor groups.

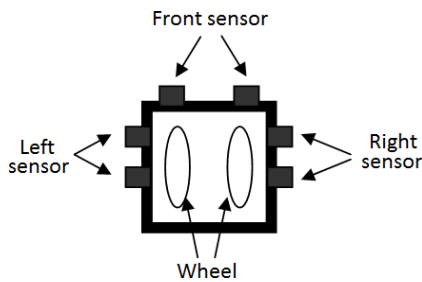


Fig. 1. Wheel and sensor arrangement of mobile robot.

B. Navigation System

The mobile robot is designed to function without reliance on a world model. The mobile robot is assumed to have some sort of radar that can detect the direction of the goal position. It shall function like a compass with its "needle" always pointing towards the north direction; in this case, the radar will be pointing towards the goal direction. It will take

one direction as a global reference to calculate its heading direction (α), and the goal direction (β). The direction the mobile robot should head towards is determined by θ , where $\theta = \beta - \alpha$, as shown in Fig. 2.

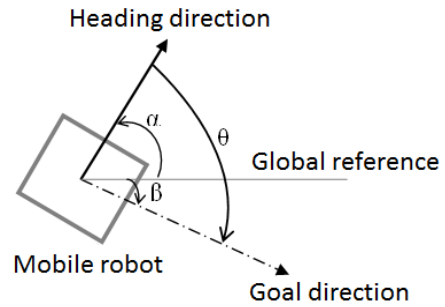


Fig. 2. Angles for navigation towards goal.

C. Hierarchical Behaviors

We designed 4 behaviors for this mobile robot in a bottom-up manner. These behaviors and the design sequence are: wandering, collision avoidance, wandering safely, and navigating towards the goal. The behaviors are of different levels in a hierarchy and classified into primitive and high-level behavior, as shown in Fig. 3.

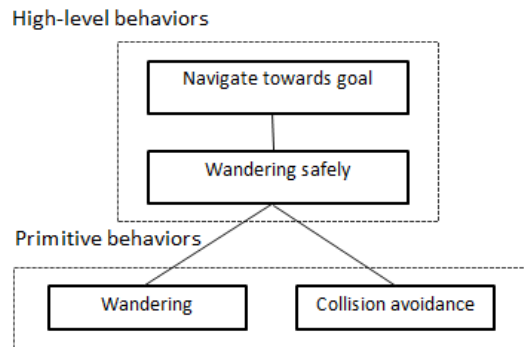


Fig. 3. Hierarchical behaviors.

Designing this system in a bottom-up manner has two advantages: 1) Robust to failure of higher-level behaviors, and 2) System is highly parallel.

In case the navigation towards goal behavior is not functioning as planned or a certain error occurs while computing the goal angle, the mobile robot will not fall into immediate danger as the system is designed to obey primitive behaviors before trying to accomplish objectives of higher-level behaviors, ensuring the survivability of the mobile robot. This is shown in the simulation result in

section III. A rule in designing bottom-up system architecture is that each level is ensured to function properly before designing the level above it. In this case, the ‘wandering safely’ behavior is ensured to be functioning before designing the behavior ‘navigating towards the goal’. In this way, while designing new behaviors for the system, we can ensure that any malfunction is solely caused by the newly designed behavior. In this way, the designed system is considered to be highly parallel.

1) Wandering

This behavior controls the direction and speed of the mobile robot. This behavior does not take any input from the sensors. The directions are: go forward, turn left, and turn right; each direction has three possible speeds: slow, medium, and fast.

2) Collision Avoidance

This behavior utilizes the sensors’ input to decide on the action to be taken. This behavior can be decomposed as follows:

- Avoid left obstacle: turn right
- Avoid right obstacle: turn left
- Avoid front obstacle: turn left/right (left/right priority)

3) Wandering Safely

This is a high-level behavior that is an emergence from the combination of the two primitive behaviors: wandering and collision avoidance. It enables the mobile robot to wander around while avoiding collision with obstacles.

4) Navigating towards the Goal

This behavior is the highest-level of behavior in this system. It takes in readings from the radar to check which direction it should head to reach the goal. It then uses the actions of ‘wandering’ to command the robot to go in the correct direction. Using fuzzy variables, this behavior can be described as follows:

- Goal on the left: turn left
- Goal on the right: turn right
- Goal in front: go forward

D. Command Fusion

As mentioned in the introduction and the section on hierarchical behaviors, each behavior will have its own decision for the mobile robot. We will make use of all of the outputs of the behaviors by performing command fusion with Mamdani fuzzy inference rules. Mamdani fuzzy inference rules are a common method used in fuzzy logic to enable a system to have linguistic variables that are similar to how humans describe things. Some of the examples of

such variables while describing distance are: very near, near, far, and very far. These variables often cannot be categorized with a clear threshold value because it is up to each individual to define how far is far etc. The reason we selected fuzzy logic is because fuzzy logic has the ability to produce a precise datum or output from such vague data. Fuzzy logic introduces a mechanism where a fixed and dynamic arbitration policy can both be implemented. This produces a smooth transition between behaviors and allows partial and concurrent activations of behaviors to be expressed. Fig. 4 shows the simple overview of the command fusion operation.

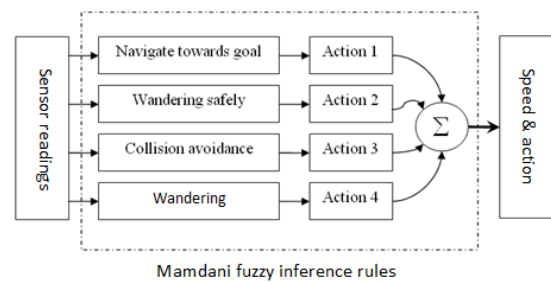


Fig. 4. Overview of command fusion.

The fusion of all the behaviors’ action is performed using the rules listed in Table 1. There are a total of 33 rules used in this system. The rules in the table can be interpreted as follow:

1. If the *left sensor* is *medium* and *front sensor* is *far* and *right sensor* is *far* and the *goal direction* is *none*, then the *action* is *forward* and *speed* is *fast*.
2. If the *left sensor* is *near* and *front sensor* is *far* and the *right sensor* is *far* and the *goal direction* is *none*, then the *action* is *forward* and *speed* is *fast*.
- ⋮
33. If the *left sensor* is *far* and *front sensor* is *far* and *right sensor* is *near* and the *goal direction* is *left*, then the *action* is *turn left 2* and *speed* is *medium*.

The rules are set with different weights. Rules for avoiding obstacles have higher weights than rules for navigating towards the goal. Hence, the obstacle avoidance rule will have more influence on the resulting action of the mobile robot when an obstacle is detected. When there is no obstacle, the rules for obstacle avoidance will not have any effect on the final outcome of the fuzzy inference system.

The same goes for moving forward, turning left, and turning right. The weights for moving forward are higher than those for turning to the left and right side. Hence, when obstacle is detected on both sides of the mobile robot, as long as the front direction has no obstacle, the mobile robot shall move in the forward direction, assuming that the goal

direction does not affect the decision making.

From the rules designed and listed in Table 1, the system we designed has 4 input variables and 2 output variables. The input variables to the fuzzy inference system are:

- Left sensor: obstacle distance from left side of the robot.
- Front sensor: obstacle distance from front of the robot.
- Right sensor: obstacle distance from right side of the robot.
- Goal direction: calculates the offset angle between robot's position and the goal's position, and indicates whether the goal is on the left, front, or right side of the mobile robot.

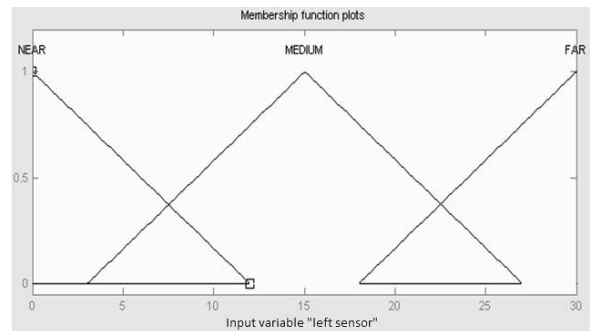
The output variables of the system are:

- Action: commands the robot to turn left, move forward, or turn right. In Table 1, turn left 2 and turn right 2 mean turning at a larger angle in that direction.
- Speed: controls the speed of the robot; slow, medium or fast

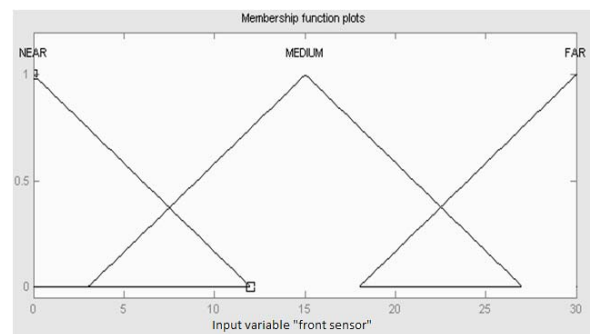
Table 1. Fuzzy inference rules

No.	Left	Front	Right	Goal direction	Action	Speed
1	Med	Far	Far	-	Forward	Fast
2	Near	Far	Far	-	Forward	Fast
3	Far	Far	Med	-	Forward	Fast
4	Med	Far	Med	-	Forward	Fast
5	Near	Far	Med	-	Forward	Fast
6	Far	Far	Near	-	Forward	Fast
7	Med	Far	Near	-	Forward	Fast
8	Near	Far	Near	-	Forward	Fast
9	Far	Med	Far	-	Turn right	Med
10	Med	Med	Far	-	Turn right	Med
11	Near	Med	Far	-	Turn right	Med
12	Far	Med	Med	-	Turn left	Med
13	Med	Med	Med	-	Turn right	Med
14	Near	Med	Med	-	Turn right	Med
15	Far	Med	Near	-	Turn left	Med
16	Med	Med	Near	-	Turn left	Med
17	Near	Med	Near	-	Forward	Med
18	Far	Near	Far	-	Turn right 2	Slow
19	Med	Near	Far	-	Turn right 2	Slow
20	Near	Near	Far	-	Turn right 2	Slow
21	Far	Near	Med	-	Turn left 2	Slow
22	Med	Near	Med	-	Turn right 2	Slow
23	Near	Near	Med	-	Turn right 2	Slow
24	Far	Near	Near	-	Turn left 2	Slow
25	Med	Near	Near	-	Turn left 2	Slow
26	Near	Near	Near	-	Turn right 2	Slow
27	Far	Far	Far	Left	Turn left	Med
28	Far	Far	Far	Front	Forward	Med
29	Far	Far	Far	Right	Turn right	Med
30	Far	-	-	Left	Turn left	Med
31	-	-	Far	Right	Turn right	Med
32	Near	Far	Far	Right	Turn right 2	Med
33	Far	Far	Near	Left	Turn left 2	Med

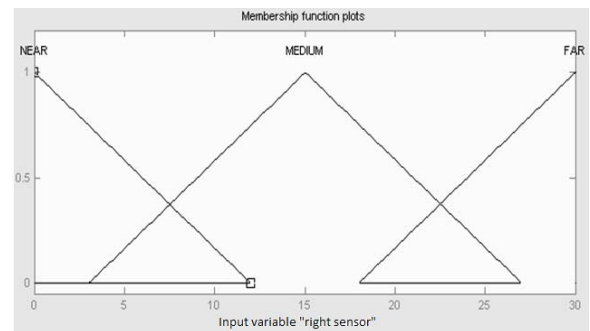
III. SIMULATION RESULT AND DISCUSSION



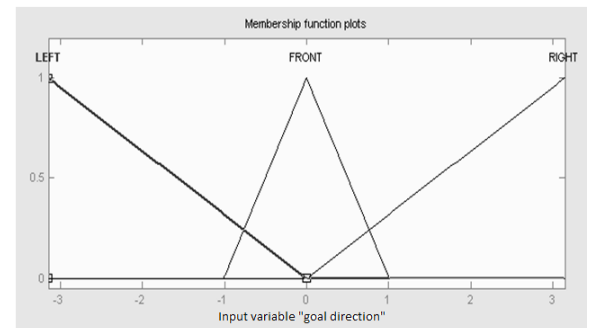
(a)



(b)



(c)



(d)

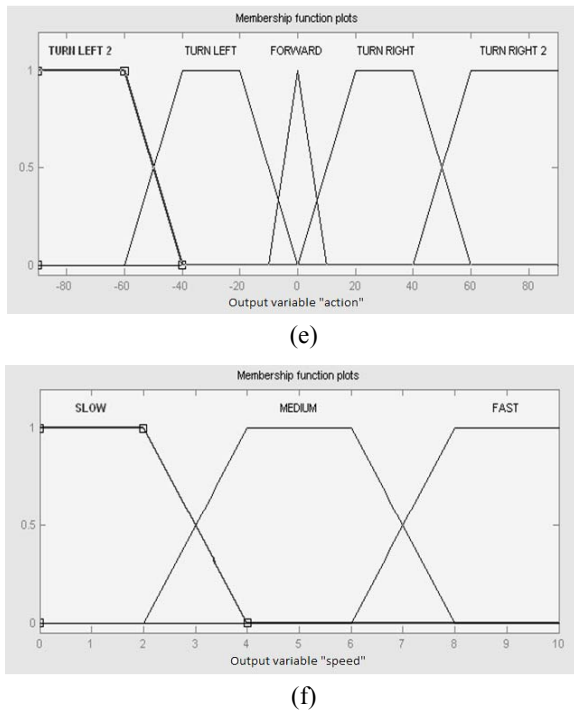


Fig. 5. Membership function plots. (a) Input variable: left sensor, (b) input variable: front sensor, (c) input variable: right sensor, (d) input variable: goal direction, (e) output variable: action, and (f) output variable: speed.

We performed the simulations in MATLAB (MathWorks, Natick, MA, USA). A user interface was created for the user to select the starting and goal position of the mobile robot. The amount, size, and position of obstacles are also configurable by the user. The membership function plots of the fuzzy system are shown in Fig. 5.

Fig. 6a showed the simulation of the navigation of the mobile robot from the starting position towards the goal position in an environment without any obstacles. As we mentioned earlier, the navigation system uses only the directional information of the goal position and the mobile robot is unaware of its position in the world or the exact location of the goal position. The system will only halt when the robot reaches the goal position or when the time limit set by the user is reached.

In Fig. 6b, it is shown that the mobile robot is able to reach the goal position by avoiding the obstacle. We can see that once the mobile robot is at a distance away from the wall, it immediately tries to head in the direction of the goal position. When the mobile robot detects the presence of the wall, it immediately distances itself from the wall. This process is repeated till the mobile robot accomplishes its objective.

In Fig. 6c, the mobile robot navigated into a dead-end. It got stuck inside for a number of steps before escaping

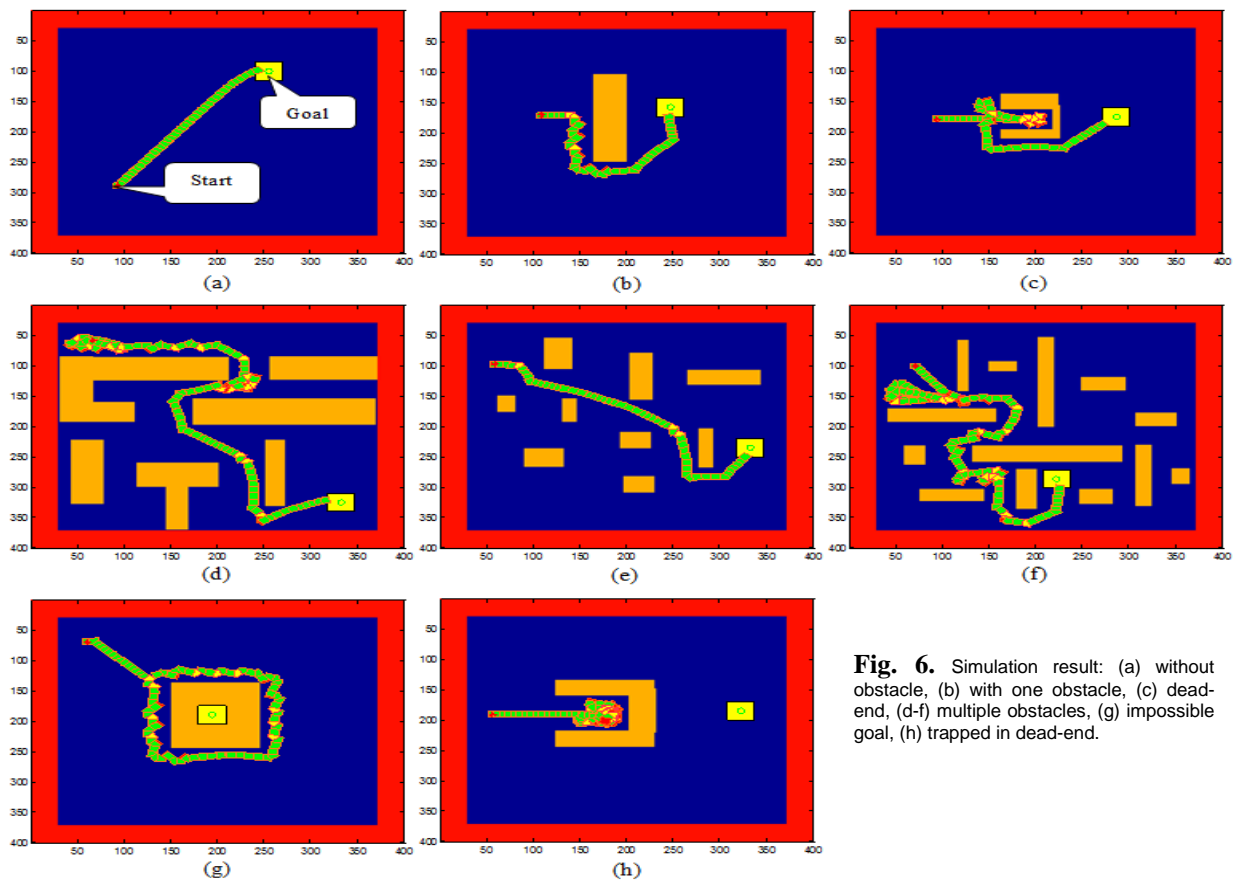


Fig. 6. Simulation result: (a) without obstacle, (b) with one obstacle, (c) dead-end, (d-f) multiple obstacles, (g) impossible goal, (h) trapped in dead-end.

from the dead-end and heading towards the goal position. This is all done by just using the 4 simple behaviors defined in section II. Various previous works by other researchers have required the system to rely on a world model and a specific behavior to escape from this situation. Our simulation showed that with the right combinations of behaviors and rules, a behavior-based system can also achieve the objective that is usually tackled by deliberative systems. We noticed that there are some places where the mobile robot got “confused” and took a longer time to reach the goal position. This is actually a trait of behavior-based systems as they often mimic biological life, and a biological organism will often get confused when it is situated in a maze without knowledge of its position or the maze’s mapping.

Fig. 6d-f further confirm the capability of a behavior-based system in performing tasks that usually require cognitive skills and dependence on the internal state representation of the real-world. These figures show that the mobile robot is able to find its way towards the goal position through a complicated course with many obstacles.

Fig. 6g demonstrates the robustness of a behavior-based system. The goal position is selected to be in an obstacle. The mobile robot is shown attempting to head towards the goal position whenever possible but the primitive behavior, collision avoidance, keeps the mobile robot out of danger whenever it gets too close to the wall. Often in a deliberative system, when such conflicts arise, the system will go berserk if there are no rules to check for those conditions. A behavior-based system has an advantage in this because if a higher-level behavior’s decision clashes with the primitive behaviors, the primitive behaviors are given higher priority over the control of the mobile robot, thus ensuring the safety and survivability of the robot in unforeseen circumstances. Fig. 6h shows that without representation of the world model or complex deliberative skills, a behavior-based system can sometimes get stuck in a dead-end. This is the trade-off for being a purely reactive system.

Based on the simulation results, we have shown that contrary to a general audience’s perception of intelligence, a system does not necessarily need to be equipped with highly deliberative skills, nor does it require representation of the real world as an internal state, in order to be able to demonstrate a certain level of intelligence. Although the system we designed lacks in spatial recognition and planning ability, it could still perform as if it had all those skills. This is all due to the advantage of command fusion of all of the behaviors’ output. A behavior alone might not prove to be useful; however, through combining all the behaviors’ decisions, an intelligent decision can be made. We are not claiming that there is no need for deliberative or spatial recognition skills in artificial intelligence, as there

are also certain drawbacks to behavior-based systems, as shown earlier.

Besides that, the robustness of behavior-based systems in addressing failure in higher-level behaviors might not be something that can be easily achieved in a deliberative system. Often when an undefined situation occurs or when there are certain faults in one small part of the system, the functionality and safety of the system will be at risk.

The characteristic of parallelism in behavior-based systems is also a great challenge to achieve in designing other systems. In a behavior-based system, as long as the bottom layer behaviors can function as they are, new behaviors can then be added. However, the same cannot be said for other systems, as most of the time, a change in one simple function of the complex system will affect the overall functionality of the whole system; much less adding an altogether new function.

We hope that through our system, we demonstrated the characteristics and advantages of a behavior-based system.

IV. CONCLUSIONS

We presented a strict behavior-based mobile robot system designed hierarchically from the bottom-up. We identified a behavior-based system to be:

- Highly reactive
- Relying minimally on real-world representation
- Highly parallel

Through each reactive behavior and the fusion of those impulsive actions with fuzzy inference rules, a behavior-based system is shown to be able to perform intelligent tasks. All these are done without having a virtual representation of the world model. The importance and advantage of parallelism demonstrated by a behavior-based system are also presented.

ACKNOWLEDGMENTS

This research was supported by Dongseo University's Ubiquitous Appliance Regional Innovation Center research grants from the Ministry of Knowledge Economy of the government of the Republic of Korea (No. B0008352).

REFERENCES

- [1] R. A. Brooks, “A robust layered control system for a mobile robot,” Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge: MA, Technical Report AIM-864, 1985.
- [2] M. Norouzi, A. Karambakhsh, M. Namazifar, and B. Savkovic,

- “Object based navigation of mobile robot with obstacle avoidance using fuzzy controller,” *Proceedings of IEEE International Conference on Control and Automation*, Christchurch, New Zealand, pp. 169-174, 2009.
- [3] S. Berman, M. A. A. de Oliveira, Y. Edan, and M. Jamshidi, “Hierarchical fuzzy behavior-based control of a multi-agent robotic system,” *Proceedings of the 7th Mediterranean Conference on Control and Automation*, Haifa, Israel, pp. 2024-2032, 1999.
- [4] A. Safiotti, “Fuzzy logic in autonomous robotics: behavior coordination,” *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, pp. 573-578, 1997.
- [5] E. Tunstel, T. Lippincott, and M. Jamshidi, “Behavior hierarchy for autonomous mobile robots: fuzzy-behavior modulation and evolution,” *International Journal of Intelligent Automation and Soft Computing*, vol. 3, no. 1, pp. 37-49, 1997.
- [6] X. Shi, P. J. A. Lever, and F. Y. Wang, “Fuzzy behavior integration and action fusion for robotic excavation,” *IEEE Transactions on Industrial Electronics*, vol. 43, no. 3, pp. 395-402, 1996.
- [7] B. B. Werger, “Cooperation without deliberation: a minimal behavior-based approach to multi-robot teams,” *Artificial Intelligence*, vol. 110, no. 2, pp. 293-320, 1999.
- [8] L. Steels, “The artificial life roots of artificial intelligence,” *Artificial Life*, vol. 1, no. 1-2, pp. 75-110, 1994.
- [9] R. A. Brooks, “Intelligence without reason,” Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge: MA, Technical Report AIM-1293, 1991.
- [10] R. A. Brooks, “Intelligence without representation,” *Artificial Intelligence*, vol. 47, no. 1-3, pp. 139-159, 1991.
- [11] B. B. Werger, M. S. Fontan, D. Goldberg, G. Hornby, M. Mataric, and S. Song, “Multiple agents from the bottom up: the interaction lab’s robot competition effort,” *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*, Providence, RI, pp. 802-803, 1997.
- [12] D. Wang, Y. Zhang, and W. Si, “Behavior-based hierarchical fuzzy control for mobile robot navigation in dynamic environment,” *Proceedings of Chinese Control and Decision Conference*, Mianyang, China, pp. 2419-2424, 2011.
- [13] F. Michaud, “Selecting behaviors using fuzzy logic,” *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, pp. 585-592, 1997.



Gan Zhen Ye

is a Masters student in Computer Engineering in Dongseo University, Busan, Korea. He received his B.S. in Electronics majoring in Robotics and Automation from Multimedia University in Malaysia (2011). His current interests include robotics, artificial intelligence, and metaheuristics.



Dae-Ki Kang

is an assistant professor at Dongseo University in Korea. He was a senior member of the engineering staff at the associated Institute of Electronics & Telecommunications Research in South Korea. He earned a PhD in computer science from Iowa State University in 2006. His research interests include social network services, machine learning, relational learning, statistical graphical models, metaheuristics, ontology learning, the Tower of Hanoi, multimedia systems, intrusion detection, Web firewall, and computer vision. Prior to joining Iowa State, he worked at two Bay-area startup companies and at Electronics and Telecommunication Research Institute in South Korea. He received a Master of Science degree in computer science from Sogang University in 1994 and a Bachelor of Engineering (BE) degree in computer science and engineering from Hanyang University in 1992.