INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

# A Fuzzy Obstacle Avoidance Controller Using a Lookup-Table Sharing Method and Its Applications for Mobile Robots

Regular Paper

Jinwook Kim, Yoon-Gu Kim, and Jinung An*

Daegu Gyeoungbuk Institute of Science & Technology, Korea
* Corresponding author E-mail: robot@dgist.ac.kr

Received 09 June 2011; Accepted 03 Oct 2011

Abstract A Lookup-Table (LUT) based design enhances the processing speed of a fuzzy obstacle avoidance controller by reducing the operation time. Also, a LUT sharing method provides efficient ways of reducing the LUT memory size. In order to share the LUT which is used for a fuzzy obstacle avoidance controller, an idea of using a basis function is developed. As applications of the shared LUT-based fuzzy controller, a laser-sensor-based fuzzy controller and an ultrasonic-sensor-based fuzzy controller are introduced in this paper. This paper suggests a LUT sharing method that reduces the LUT buffer size without a significant degradation of the performance. The LUT sharing method makes the buffer size independent of the fuzzy system's complexity. A simulation using MSRDS (Microsoft Robotics Developer Studio) is used to evaluate the proposed method. To investigate the performance of the controller, experiments are carried out using a Pioneer P3-DX with LabVIEW as an integration tool. Although the simulation and experiments show little difference between the fully valued LUT-based method and the LUT sharing method in terms of the operation time, the LUT sharing method reduces almost 95% of the full-valued LUT-based buffer size.

Keywords Autonomous Navigation, Obstacle Avoidance, Fuzzy Controller, Mobile Robot, Lookup Table.

## 1. Introduction

Autonomous navigation algorithms are applied to various applications for mobile robots to drive toward their target position without a remote control program. In the mapping step of an autonomous navigation application, the robot acquires the location and contour information of obstacles from sensors and updates its navigation map. A navigation path from the current position to the target position on the map is determined by the updated map data. If there are no changes in the map during the robot navigation process, obstacle avoidance is not necessary because a well-defined navigation path will lead the robot to the target position safely. Even in the event that unexpected obstacles suddenly appear on the navigation path, the robot is required to perform obstacle avoidance in order to navigate the planned path without colliding with these types of obstacles. To escape the unexpected obstacles and continue toward the target position under real-time navigation, the robot should determine its steering direction and velocity instantly. This paper is focused on performance improvement and implemental optimization for a fast real-time obstacle avoidance system of robot navigation.

There are several algorithms pertaining to robot obstacle avoidance, including potential field algorithms [1], vector field histogram algorithms [2], and fuzzy controller algorithms [3]. The fundamentals of fuzzy controller methods were presented by Wang [4], Palm [5], Passino [6], Aguirre [7], and others [8, 9]. As a result of these papers, fuzzy controller methods have more advantages in real-time applications because they provide simpler and more intuitive methods of robot obstacle avoidance. This is why we chose fuzzy controller methods for obstacle avoidance here. Mbede [10] proposed a neuro-fuzzy motion controller, and Lilly [11] presented a P/N fuzzy obstacle avoidance controller. Lilly's work provided a more intuitive method for robot obstacle avoidance through its use of negative fuzzy rules to eliminate redundant fuzzy rules. Kiendl [12] and Branson [13] also presented methods of utilizing negative rules in fuzzy systems. The fuzzy controller in this paper is based on the P/N rule fuzzy controller method proposed in Lilly's paper [11]. We verified the performance of the previous P/N rule fuzzy controller method using a robotics simulator in MSRDS (Microsoft Robotics Development Studio) 2008 [14]. We also devised an improved algorithm based on the previous P/N rule fuzzy controllers, verified them in a simulator, and experimented in a real environment.

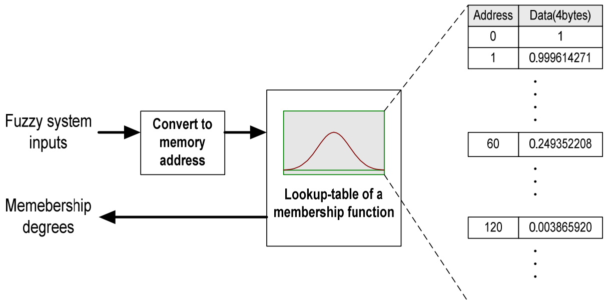
In Section 2, the fundamentals of the P/N rule fuzzy controller and the full LUT-based design of the P/N fuzzy controller are introduced. The proposed LUT sharing method for the controller is described in Section 3. Section 4 and Section 5 present the laser range-finder-based 50-rule P/N fuzzy controller and the ultrasonic-sensor-based 18-rule P/N fuzzy controller, respectively. Simulation and experimentation results are shown in Section 6. Finally, concluding remarks are summarized in Section 7.

## 2. A Positive/Negative rule fuzzy controller and its full-LUT-based design

### *2.1 A Positive/Negative rule fuzzy controller*

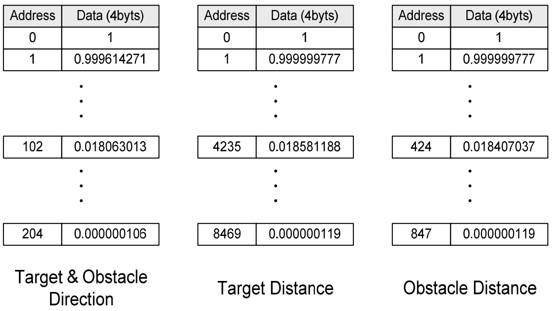
Fuzzy rules of traditional fuzzy systems only describe the operations that are to be run. In Positive/Negative (P/N) rule fuzzy system theory, the concept of negative fuzzy rules was introduced to prescribe actions to avoid rather than execute. Two types of rules, positive rules and negative rules, are used in a P/N rule fuzzy controller. Positive-rules derive the proper output of the P/N-rule-based fuzzy system, comparable to typical fuzzy system rules. In contrast, negative rules cause the fuzzy system not to produce improper fuzzy output. The P/N rule fuzzy system can be applied to obstacle avoidance for the autonomous navigation of mobile robots. Positive rules direct the robot to a target place and negative rules prevent the robot from colliding with obstacles. The effect of positive rules is dominant in a non-obstacle environment because negative rules begin to affect navigation only when obstacles appear on the navigation path of the robot. If the detected obstacles are far from the robot, negative rules rarely affect the system. As the obstacles come closer to the robot, the negative rules dominate the system.

### *2.2 Full-LUT-based design of a Positive/Negative rule fuzzy controller*


Fuzzy system inputs
Convert to memory address
Lookup-table of a membership function
Memebership degrees

| Address | Data(4bytes) |
|---|---|
| 0 | 1 |
| 1 | 0.999614271 |
| ⋮ | |
| 60 | 0.249352208 |
| ⋮ | |
| 120 | 0.003865920 |
| ⋮ | |

**Figure 1.** Fuzzy membership degree acquisition on lookup-table based architecture



Target & Obstacle Direction

| Address | Data (4byts) |
|---|---|
| 0 | 1 |
| 1 | 0.999614271 |
| ⋮ | |
| 102 | 0.018063013 |
| ⋮ | |
| 204 | 0.000000106 |

Target Distance

| Address | Data (4byts) |
|---|---|
| 0 | 1 |
| 1 | 0.999999777 |
| ⋮ | |
| 4235 | 0.018581188 |
| ⋮ | |
| 8469 | 0.000000119 |

Obstacle Distance

| Address | Data (4byts) |
|---|---|
| 0 | 1 |
| 1 | 0.999999777 |
| ⋮ | |
| 424 | 0.018407037 |
| ⋮ | |
| 847 | 0.000000119 |

**Figure 2.** Lookup-table (LUT) buffers of full-LUT-based architecture

For secure navigation for mobile robots, the computation delay of the obstacle avoidance controller is one of the most important factors. To reduce the delay, we decided to reduce the computational complexity of the controller. It is not to be denied that faster decision when avoiding obstacles contributes to more accurate avoidance. In our design, we adopted a P/N fuzzy controller method which is simpler and more intuitive for robot obstacle avoidance. However, fuzzy membership functions formed as Gaussian distribution curves are the most critical problem related to fast operations because the fuzzy controller should process numerous floating point operations to calculate the fuzzy membership degrees. The computations for fuzzy membership degrees could be a burden that causes difficulties with real-time navigation. Therefore, we attempted to reduce the computations for membership degrees through the use of the LUTs of membership functions. The basic idea is that the fuzzy controller does not need to calculate the membership function degree upon the receipt of sensor data each time, as all membership functions and their each output values corresponding to their input values are previously determined according to the characteristics of the function. Therefore, it is possible to store all membership function data in the LUTs in advance and search for membership degrees from the LUTs without any computational time. Fig. 2 illustrates the process of obtaining fuzzy membership degrees from the LUT. The LUT buffers of all fuzzy membership functions are shown in Fig. 3. If the fuzzy controller obtains each membership degree from LUT buffers, the robot can reduce the processing time and perform more frequent obstacle-avoidance operations because the operation delay in the multiplication of the floating point operands is eliminated.

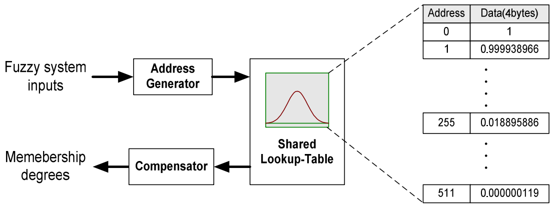


**Figure 3.** Fuzzy membership degree acquisition using a shared lookup-table

## 3. Lookup-Table sharing method

It is quite attractive to use LUTs that are capable of rapid operations in a fuzzy controller, but there are also disadvantages in a LUT-based design. To maintain the LUTs of membership functions, the fuzzy controller should guarantee a sufficient size of the buffer memory for the LUTs. Because a large LUT memory is a burden on the system, we tried to reduce the memory through the use of a shared LUT. The shared LUT contains the function values of a basic Gaussian distribution function that we defined. The equation of the basis function can be expressed as follows:

$$f_{basis} = e^{-\frac{x_{trans}^2}{2\sigma_{BasisFunction}^2}} . \qquad (1)$$

The variance of the basis function, $\sigma_{BasisFunction}$, is set to 128 for the 50-rule P/N fuzzy controller. This idea stems from the fact that all membership functions of our fuzzy controller are based on a Gaussian distribution which is characterized by the variance ($\sigma$) and the center ($c$). Therefore, each membership function's output value is simply obtained from the basis function by converting the input value $x_{input}$ into the proper converted value $x_{Converted}$, which is calculated as follows:

$$x_{Converted} = \frac{\sigma_{BasisFunction}}{\sigma_{MembershipFunction}}(x_{input} - c_{MembershipFunction}) . \qquad (2)$$

The input data from a laser range finder (LRF) are discrete and are converted into integer values so that they can be used as the indexes of the LUTs. The shared LUT is also indexed by integral numbers to simplify the use of buffers. Because the calculation result, $x_{Converted}$, can contain a decimal fraction value, it is not possible to provide the output value of the original membership function using the shared LUT because it provides only the output values of integral inputs. Although we cannot use non-integer indexes, we can obtain an approximate value by rounding off $x_{Converted}$ to its nearest integer.

As shown above, $f_{basis}$ is a Gaussian distribution function; hence, the left part of the basis function in the shared LUT can be eliminated to remove redundancy. The absolute values of the inputs are used to generate an indexing address corresponding to the left part of the basis function. Finally, the indexing address of the shared LUT is calculated as follows:

$$Address_{LUT} = \left| round(\frac{\sigma_{BasisFunction}}{\sigma_{MembershipFunction}}(x_{input} - c_{MembershipFunction}) \right| . \qquad (3)$$

Fig. 3 shows a diagram of the fuzzy membership degree acquisition process using the LUT sharing method. The *Address Generator* generates the indexing address of the shared LUT, *Address*$_{LUT}$; the calculated address is proper for the characteristics of each of its membership function. In each indexing address, 32-bit floating point data is stored. Also, the range of indexing address is between 0 and 511, as shown in Fig. 3. If the calculated address, *Address*$_{LUT}$, exceeds a length of 511, we change the value to 511 and the LUT releases 0 as the output. This also indicates that we consider output values below 0.000000119 as 0 in order to reduce the buffer size.

After reading the output of the function from the shared LUT, *Compensator* compensates for the output value, as it is not identical to an expected membership degree. However, this may not be necessary when the errors between the shared LUT outputs and the original output values are too small to affect the navigation performance. We confirmed this in simulations and determined it was feasible to eliminate *Compensator* from the design.

## 4. 50-Rule P/N Fuzzy Controller for Laser-Range-Finder-based Obstacle Avoidance

### *4.1 Design of 50-Rule P/N Fuzzy Controller*

The 50-rule P/N fuzzy controller is designed with two types of input parameters, the first is sensor data from a laser range finder and the second is the relative target position based on the current position of the robot. Fig. 4 shows the measurement data from the laser range finder. The laser range finder sensor provides the distance vectors of directions over 180 degree with an angular resolution of 0.5 degrees. To simplify the fuzzy system, we converted the scale of inputs into integer values by multiplying the input value by 2. Once the target position is determined, the position is a constant on the navigation map. However, the relative target position varies according to changes of the current robot position. As the robot moves, the robot controller should update the relative target position, which consists of a target distance and a target direction. The output of the fuzzy controller is the steering direction of the robot. In a real application, however, the controller converts the direction information into values for steering, as in the velocities of two wheels.

In the 50 P/N fuzzy rules, 25 positive rules are related to the target position and 25 negative rules are related to the positions of multiple obstacles. The fuzzification of the 50-rule P/N fuzzy controller for the laser-range-finder-based obstacle avoidance method is explained in the next section.

### *4.2 Fuzzification*

Fuzzification of the input is described by the membership functions shown in Fig. 5. Each membership function of a fuzzy input is a Gaussian distribution function. These are expressed as Eq. (4)

$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \tag{4}$$

where $c$ denotes the center of the distribution and $\sigma$ denotes the variance of the function. Table 1 represents the $c$ and $\sigma$ values of each fuzzy membership function. The fuzzy sets Hard Left (HL), Soft Left (SL), Straight (S), Soft Right (SR), and Hard Right (HR) are for the fuzzy inputs of the target or the obstacle direction. In addition, Zero (Z), Very Near (VN), Near (N), Far (Far), and Very Far (VF) are related to the fuzzy inputs of the target or the obstacle distance. The output fuzzy sets Hard Left (HL), Left (L), Soft Left (SL), Straight (S), Soft Right (SR), Right (R), and Hard Right (HR) are for positive rules, while $\overline{HL}$, $\overline{L}$, $\overline{SL}$, $\overline{S}$, $\overline{SR}$, $\overline{R}$, and $\overline{HR}$ are for negative rules.

### *4.3 Rule Base*

The expert-provided positive rule base is shown in Table 2 and the negative rule base is given in Table 3. The positive rule base can be represented by the following rules:

*Rule 1P*
If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is Z then y is HL
*Rule 2P*
If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is VN then y is HL
*Rule 3P*
If $x_{TargetDirection}$ is HL and $x_{TargetDistance}$ is N then y is L

.
.
.

*Rule 25P*
If $x_{TargetDirection}$ is HR and $x_{TargetDistance}$ is VF then y is SR.

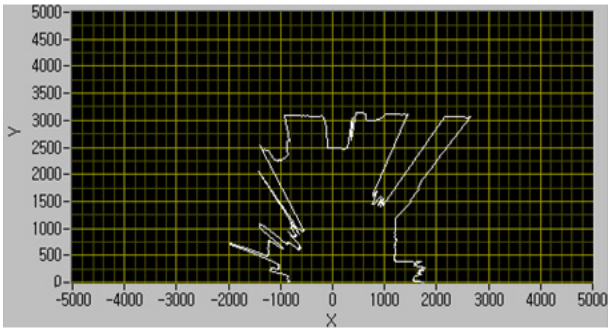


**Figure 4.** Measurement of the laser range finder

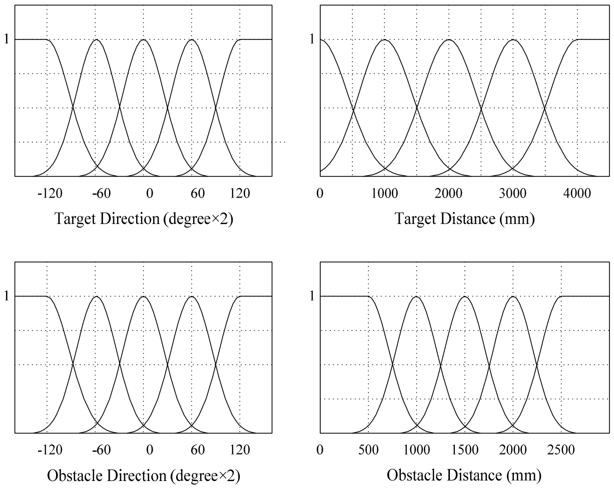


**Figure 5.** Membership functions of the designed 50-rule fuzzy controller

| | | Target | | Obstacle | |
|---|---|---|---|---|---|
| | | Center ($c$) | Variation ($\sigma$) | Center ($c$) | Variation ($\sigma$) |
| Direction (degree×2) | HL | -120 | 36 | -120 | 36 |
| | SL | -60 | 36 | -60 | 36 |
| | S | 0 | 36 | 0 | 36 |
| | SR | 60 | 36 | 60 | 36 |
| | HR | 120 | 36 | 120 | 36 |
| Distance (mm) | Z | 0 | 6000 | 500 | 300 |
| | VN | 10000 | 6000 | 1000 | 300 |
| | N | 20000 | 6000 | 1500 | 300 |
| | F | 30000 | 6000 | 2000 | 300 |
| | VF | 40000 | 6000 | 2500 | 300 |

**Table 1.** Characteristics of the Fuzzy Membership Functions of the 50-rule P/N Fuzzy Controller

| Steering Direction | | Target Direction | | | | |
|---|---|---|---|---|---|---|
| | | HL | SL | S | SR | HR |
| Target Distance | Z | HL | HL | S | HR | HR |
| | VN | HL | L | S | R | HR |
| | N | L | L | S | R | R |
| | F | L | SL | S | SR | R |
| | VF | SL | S | S | S | SR |

**Table 2.** Rule Base of the 25-rule Positive Fuzzy System (Expert-Provided)

| Steering Direction | | Obstacle Direction | | | | |
|---|---|---|---|---|---|---|
| | | HL | SL | S | SR | HR |
| Obstacle Distance | Z | $\overline{HL}$ | $\overline{HL}$ | $\overline{S}$ | $\overline{HR}$ | $\overline{HR}$ |
| | VN | $\overline{HL}$ | $\overline{L}$ | $\overline{S}$ | $\overline{R}$ | $\overline{HR}$ |
| | N | $\overline{L}$ | $\overline{L}$ | $\overline{S}$ | $\overline{R}$ | $\overline{R}$ |
| | F | $\overline{L}$ | $\overline{SL}$ | $\overline{S}$ | $\overline{SR}$ | $\overline{R}$ |
| | VF | $\overline{SL}$ | $\overline{S}$ | $\overline{S}$ | $\overline{S}$ | $\overline{SR}$ |

**Table 3.** Rule Base of the 25-rule Negative Fuzzy System (Expert-Provided)

The negative rule base in Table 3 is represented in the same manner of the above rules, as follows:

*Rule 1N*
If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is Z then y is not HL
*Rule 2N*
If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is VN then y is not HL
*Rule 3N*
If $x_{ObstacleDirection}$ is HL and $x_{ObstacleDistance}$ is N then y is not L

.

.

.

*Rule 25N*
If $x_{ObstacleDirection}$ is HR and $x_{ObstacleDistance}$ is VF then y is not SR.

| Degrees of positive-rule fuzzy sets | | Degree of negative-rule fuzzy sets | |
|---|---|---|---|
| HL | $\mu_{HL}$ | HL | $\mu_{\overline{HL}}$ |
| SL | $\mu_{SL}$ | SL | $\mu_{\overline{SL}}$ |
| S | $\mu_{S}$ | S | $\mu_{\overline{S}}$ |
| SR | $\mu_{SR}$ | SR | $\mu_{\overline{SR}}$ |
| HR | $\mu_{HR}$ | HR | $\mu_{\overline{HR}}$ |
| Z | $\mu_{Z}$ | Z | $\mu_{\overline{Z}}$ |
| VN | $\mu_{VN}$ | VN | $\mu_{\overline{VN}}$ |
| N | $\mu_{N}$ | N | $\mu_{\overline{N}}$ |
| F | $\mu_{F}$ | F | $\mu_{\overline{F}}$ |
| VF | $\mu_{VF}$ | VF | $\mu_{\overline{VF}}$ |

**Table 4.** Membership Degrees of Input Fuzzy Sets

The notation of the membership degree of each fuzzy set is expressed as shown in Table 4. The weight of the positive rule '*Rule 1P*' is determined by the multiplication of $\mu_{HL}$ and $\mu_Z$, and the weight of '*Rule 2P*' is calculated by the multiplication of $\mu_{HL}$ and $\mu_{VN}$. The same approach is applied to the other rules. The combined degree of the positive rule '*Rule 1P*' and its corresponding negative rules is calculated as follows:

$$\mu_1 = (offset + \mu_{HL}\cdot\mu_Z)\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{Z}_i}). \qquad (5)$$

Similarly, the other combined degrees are calculated as follows:

$$\mu_2 = (offset + \mu_{HL}\cdot\mu_{VN})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{Z}_i}),$$

$$\mu_3 = (offset + \mu_{HL}\cdot\mu_{N})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{N}_i}),$$

$$\mu_4 = (offset + \mu_{HL}\cdot\mu_{F})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{N}_i}),$$

$$\mu_5 = (offset + \mu_{HL}\cdot\mu_{VF})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_6 = (offset + \mu_{SL}\cdot\mu_{Z})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{Z}_i}),$$

$$\mu_7 = (offset + \mu_{SL}\cdot\mu_{VN})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{N}_i}),$$

$$\mu_8 = (offset + \mu_{SL}\cdot\mu_{N})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{N}_i}),$$

$$\mu_9 = (offset + \mu_{SL}\cdot\mu_{F})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{HL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_{10} = (offset + \mu_{SL}\cdot\mu_{VF})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{11} = (offset + \mu_{S}\cdot\mu_{Z})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{12} = (offset + \mu_{S}\cdot\mu_{VN})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_{,i}}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{13} = (offset + \mu_{S}\cdot\mu_{N})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{14} = (offset + \mu_{S}\cdot\mu_{F})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{15} = (offset + \mu_{S}\cdot\mu_{VF})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SL}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{16} = (offset + \mu_{SR}\cdot\mu_{Z})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{VN}_i}),$$

$$\mu_{17} = (offset + \mu_{SR}\cdot\mu_{VN})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_{18} = (offset + \mu_{SR}\cdot\mu_{N})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_{19} = (offset + \mu_{SR}\cdot\mu_{F})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{20} = (offset + \mu_{S}\cdot\mu_{VF})$$
$$\times\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{VF}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{S}_i}\cdot\mu_{\overline{VF}_i}),$$

$$\mu_{21} = (offset + \mu_{HR}\cdot\mu_{Z})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{VN}_i}),$$

$$\mu_{22} = (offset + \mu_{HR}\cdot\mu_{VN})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{Z}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{VN}_i}),$$

$$\mu_{23} = (offset + \mu_{HR}\cdot\mu_{N})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_{24} = (offset + \mu_{HR}\cdot\mu_{F})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{VN}_i})(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{N}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{F}_i}),$$

$$\mu_{25} = (offset + \mu_{HR}\cdot\mu_{VF})\prod_{i=1}^{\#ofObstacles}(1-\mu_{\overline{SR}_i}\cdot\mu_{\overline{F}_i})(1-\mu_{\overline{HR}_i}\cdot\mu_{\overline{VF}_i}). \qquad (6)$$

### *4.4 Defuzzification*

Finally, the output value of the 50-rule P/N fuzzy controller is calculated as the weighted average of all output fuzzy sets:

$$\begin{cases} y=\dfrac{w_{HL}\cdot HL+w_L\cdot L+w_{SL}\cdot SL+w_S\cdot S}{w_{HL}+w_L+w_{SL}+w_S} & (w_{HL}+w_L+w_{SL}\geq w_{HR}+w_R+w_{SR}), \\ y=\dfrac{w_{HR}\cdot HR+w_R\cdot R+w_{SR}\cdot SR+w_S\cdot S}{w_{HR}+w_R+w_{SR}+w_S} & (w_{HL}+w_L+w_{SL}<w_{HR}+w_R+w_{SR}) \end{cases} \quad . \tag{7}$$

The weights of the output fuzzy sets are as follows:

$$\begin{aligned} w_{HL} &= \mu_1+\mu_2+\mu_6, \\ w_L &= \mu_3+\mu_4+\mu_7+\mu_8, \\ w_{SL} &= \mu_5+\mu_9, \\ w_S &= \mu_{10}+\mu_{11}+\mu_{12}+\mu_{13}+\mu_{14}+\mu_{15}+\mu_{20}, \\ w_{SR} &= \mu_{19}+\mu_{25}, \\ w_R &= \mu_{17}+\mu_{18}+\mu_{23}+\mu_{24}, \\ w_{HR} &= \mu_{16}+\mu_{21}+\mu_{22}. \end{aligned} \tag{8}$$

| | | Target | | Obstacle | |
|---|---|---|---|---|---|
| | | Center ($c$) | Variation ($\sigma$) | Center ($c$) | Variation ($\sigma$) |
| Direction (degree×2) | HL | -120 | 36 | -120 | 72 |
| | S | 0 | 36 | 0 | 72 |
| | HR | 120 | 36 | 120 | 72 |
| Distance (mm) | Z | 0 | 1500 | 700 | 150 |
| | N | 15000 | 1500 | 1200 | 150 |
| | F | 30000 | 1500 | 1700 | 150 |

**Table 5.** Characteristics of Fuzzy Membership Functions of the 18-rule P/N Fuzzy Controller

| Steering Direction | | Target Direction | | |
|---|---|---|---|---|
| | | L | S | R |
| Target Distance | Z | HL | S | HR |
| | N | L | S | R |
| | F | SL | S | SR |

**Table 6.** Rule Base of 9-rule Positive Fuzzy System (Expert-Provided)

| Steering Direction | | Obstacle Direction | | |
|---|---|---|---|---|
| | | SL | S | SR |
| Obstacle Distance | Z | $\overline{HL}$ | $\overline{S}$ | $\overline{HR}$ |
| | N | $\overline{L}$ | $\overline{S}$ | $\overline{R}$ |
| | F | $\overline{SL}$ | $\overline{S}$ | $\overline{SR}$ |

**Table 7.** Rule Base of the 9-rule Negative Fuzzy System (Expert-Provided)

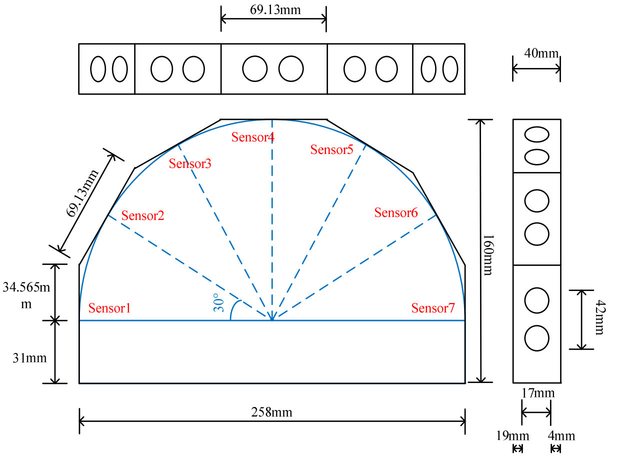


**Figure 6.** Implementation of the ultrasonic sensors for the 18-rule fuzzy obstacle avoidance fuzzy controller

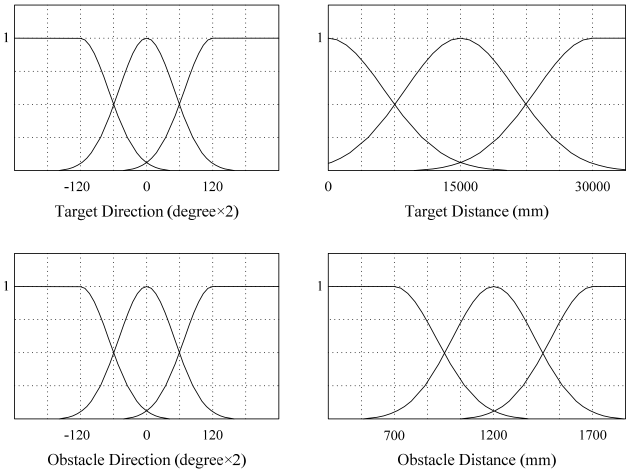


**Figure 7.** Membership functions of the designed 18-rule fuzzy controller

## 5. 18-Rule P/N Fuzzy Controller for Ultra-sonic Sensor based Obstacle Avoidance

### *5.1 Design of 18-Rule P/N Fuzzy Controller*

The 18-rule P/N fuzzy controller is designed as an ultrasonic sensor based obstacle avoidance system for mobile robots. Unlike the laser range finder, ultra-sonic sensors implemented here have an angular resolution of 30° and are arranged with 7 sensors, as shown in Fig. 6. Each sensor detects the nearest obstacle in front of the sensor within a distance from 25mm to 3m using a 30° angular range. Because its angular resolution is relatively low compared to a laser range finder, we designed a simpler fuzzy controller based on 18 rules. In a manner similar to that of the 50-rule P/N fuzzy controller, 9 positive rules are related to a target position and 9 negative rules are related to the positions of multiple obstacles.

### 5.2 Fuzzification

Fuzzification of the 18-rule P/N fuzzy controller can be described by the membership functions shown in Fig. 7. Each membership function is a Gaussian distribution function. Table 5 represents the $c$ and $\sigma$ values of each fuzzy membership function. The fuzzy sets, Left (L), Straight (S), Right (R), are for the fuzzy inputs of the target or the obstacle direction. Also, Zero (Z), Near (N), and Far (Far) are related to the fuzzy inputs of the target or the obstacle distance. The output fuzzy sets Hard Left (HL), Left (L), Soft Left (SL), Straight (S), Soft Right (SR), Right (R), and Hard Right (HR) are for positive rules, while $\overline{HL}$, $\overline{L}$, $\overline{SL}$, $\overline{S}$, $\overline{SR}$, $\overline{R}$, and $\overline{HR}$ are for negative rules.

### 5.3 Rule Base

Each rule base of positive and negative rules is shown in Table 6 and Table 7, respectively. The positive rule base can be represented by following rules:

<u>*Rule 1P*</u>
If $x_{\text{TargetDirection}}$ is L and $x_{\text{TargetDistance}}$ is Z then y is HL
<u>*Rule 2P*</u>
If $x_{\text{TargetDirection}}$ is L and $x_{\text{TargetDistance}}$ is N then y is L
<u>*Rule 3P*</u>
If $x_{\text{TargetDirection}}$ is L and $x_{\text{TargetDistance}}$ is F then y is SL
<u>*Rule4P*</u>
If $x_{\text{TargetDirection}}$ is S and $x_{\text{TargetDistance}}$ is Z then y is S
<u>*Rule 5P*</u>
If $x_{\text{TargetDirection}}$ is S and $x_{\text{TargetDistance}}$ is N then y is S
<u>*Rule 6P*</u>
If $x_{\text{TargetDirection}}$ is S and $x_{\text{TargetDistance}}$ is F then y is S
<u>*Rule7P*</u>
If $x_{\text{TargetDirection}}$ is R and $x_{\text{TargetDistance}}$ is Z then y is HR
<u>*Rule 8P*</u>
If $x_{\text{TargetDirection}}$ is R and $x_{\text{TargetDistance}}$ is N then y is R
<u>*Rule 9P*</u>
If $x_{\text{TargetDirection}}$ is R and $x_{\text{TargetDistance}}$ is F then y is SR

The negative rule base in Table 7 is also represented as follows:

<u>*Rule 1N*</u>
If $x_{\text{ObstacleDirection}}$ is L and $x_{\text{ObstacleDistance}}$ is Z then y is not HL
<u>*Rule 2N*</u>
If $x_{\text{ObstacleDirection}}$ is L and $x_{\text{ObstacleDistance}}$ is N then y is not L

.
.
.

<u>*Rule 9N*</u>
If $x_{\text{ObstacleDirection}}$ is R and $x_{\text{ObstacleDistance}}$ is F then y is not SR.

The combined degrees of the 9 positive rules are calculated as follows:

$$\mu_1 = (offset + \mu_L \cdot \mu_Z) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{L}_i} \cdot \mu_{\overline{Z}_i}),$$

$$\mu_2 = (offset + \mu_L \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{L}_i} \cdot \mu_{\overline{N}_i}),$$

$$\begin{aligned} \mu_3 &= (offset + \mu_L \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{L}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_4 &= (offset + \mu_S \cdot \mu_Z) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{Z}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{N}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_5 &= (offset + \mu_S \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{Z}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{N}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_6 &= (offset + \mu_S \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{Z}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{N}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_7 &= (offset + \mu_R \cdot \mu_Z) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{R}_i} \cdot \mu_{\overline{Z}_i}), \\ \mu_8 &= (offset + \mu_R \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{R}_i} \cdot \mu_{\overline{N}_i}), \\ \mu_9 &= (offset + \mu_R \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{R}_i} \cdot \mu_{\overline{F}_i}). \end{aligned} \tag{9}$$

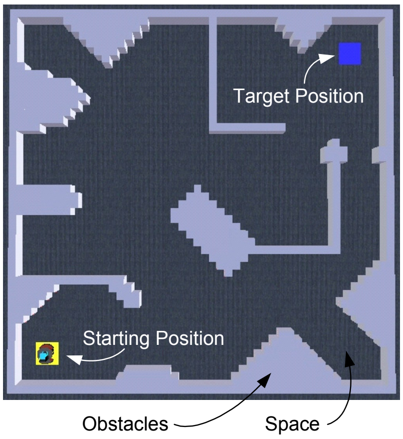


**Figure 8.** Simulation map of MSRDS (Microsoft Robotics Developer Studio)

### 5.4 Defuzzification

The output value of the 18-rule P/N fuzzy controller is also calculated as the weighted average of all output fuzzy sets:

$$\begin{cases} y = \dfrac{w_{HL} \cdot HL + w_L \cdot L + w_{SL} \cdot SL + w_S \cdot S}{w_{HL} + w_L + w_{SL} + w_S} & (w_{HL} + w_L + w_{SL} \geq w_{HR} + w_R + w_{SR}), \\ y = \dfrac{w_{HR} \cdot HR + w_R \cdot R + w_{SR} \cdot SR + w_S \cdot S}{w_{HR} + w_R + w_{SR} + w_S} & (w_{HL} + w_L + w_{SL} < w_{HR} + w_R + w_{SR}) \end{cases} . \tag{10}$$

The weights of output fuzzy sets are as follows:

$$\begin{aligned} w_{HL} &= \mu_1, \\ w_L &= \mu_2, \\ w_{SL} &= \mu_3, \\ w_S &= \mu_4 + \mu_5 + \mu_6, \\ w_{SR} &= \mu_7, \\ w_R &= \mu_8, \\ w_{HR} &= \mu_9. \end{aligned} \tag{11}$$

## 6. Simulation and Experiments

A simulation of the designed 50-rule P/N fuzzy controller was performed on MSRDS (Microsoft Robotics Developer Studio). The simulation was performed using an Intel® Core™ 2 Quad 2.4GHz CPU and 3.24GB RAM computer system. Fig. 8 shows a robot navigation map of the graphic simulator. A simulation robot navigates from the starting position to the target position on the map. If there are no obstacles in the map, the robot sets its path as a straight line from the starting position to the target position. If there are obstacles on the navigation path, the robot must change and update its path to avoid the obstacles. To evaluate the proposed method on the simulator, we created a service module for each fuzzy controller method and inserted it into the MSRDS Visual Programming Language (VPL) diagram. The service modules were programmed in the C# programming language and created by the Microsoft Visual Studio compiler. The sensor modules and driving service modules of the graphic simulator were imported using the service libraries of MSRDS. Because a graphic simulation environment requires high-performance processors, the simulation was performed on a high-performance computer system. In this case, the target systems of our method are mobile robot systems. Therefore, we used high-performance computer systems that may not be applicable to mobile embedded systems only for the graphical simulation. We also considered the processing time only for the fuzzy operation because the simulation system uses most of the processing time for graphical operations.

Table 8 presents the implementation results of each design. The buffer size of the full-LUT shown in Fig. 2 is (204+8469+847) × 4 bytes, while that of the shared LUT presented in Fig. 3 is 512 × 4 bytes. The processing clock cycle of each design in the table is the average computation of the clock cycles used in the avoidance operations during the navigation process from the starting position to the target position on the map. A fuzzy controller without LUT does not require a buffer for the LUT(s), but it does need 8 times as much time for the fuzzy operation compared to the full-LUT-based fuzzy controller. If all membership functions have LUTs, the operation time can be greatly reduced. However, this requires about 38Kbytes of LUT memory. The simulation result for the LUT sharing method showed that the fuzzy controller can operate well only with 5.34% of the full LUT's memory size. Although the controller requires about 1.5 times more operation time than the full-LUT-based design, the comparative reduction of the memory size is more advantageous for the system. In brief, the simulation results show that the LUT-based fuzzy controller greatly reduces the computational latency and obviously enhances the performance of a real-time obstacle avoidance system for an autonomous robot. Also, the LUT sharing method offers an approach that uses LUT-based architecture without the additional expense of added buffer memory. Figs. 9, 10, and 12 show the results, which were verified in the experiments based on a mobile robot (Pioneer P3-DX). Figs. 11 and 13 show the experiment processes in a cluttered environment for the result shown in Figs. 10 and 12, respectively.

| | Calculating w/o a LUT | Using the full LUT | Sharing a LUT |
|---|---|---|---|
| Buffer size of the lookup table (bytes) | 0 | 38,080 | 2,048 |
| Clock cycles per 1 avoidance (cycles) | 427,390 | 84,797 | 127,494 |

**Table 8.** Buffer size and processing cycles of each fuzzy controller

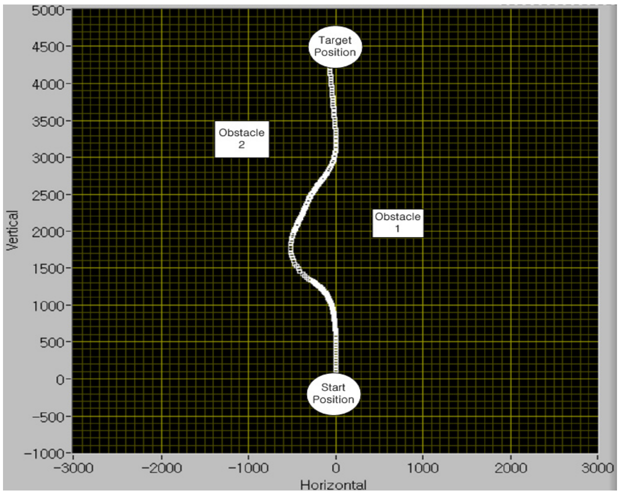


**Figure 9.** Fuzzy controller implementation for obstacle avoidance with the LabVIEW application on a mobile robot (Pioneer P3-DX)

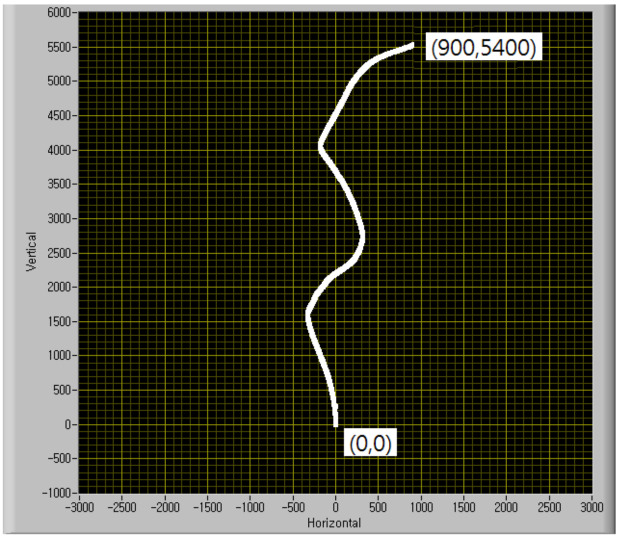


**Figure 10.** Experimental result for obstacle avoidance while navigating from the starting position (0, 0) to the target position (900, 5400) in a cluttered environment

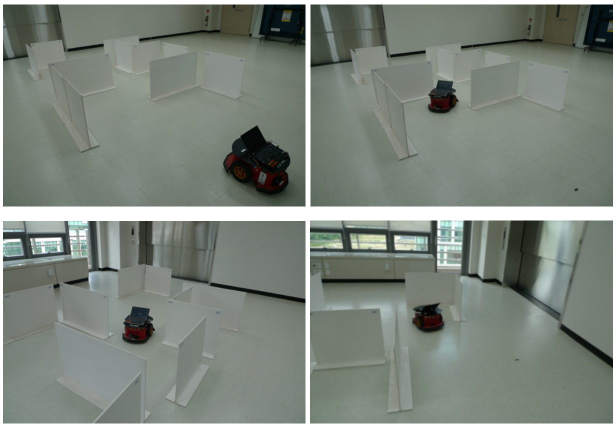


**Figure 11.** Experiment process in a cluttered environment for the result shown in Fig. 10

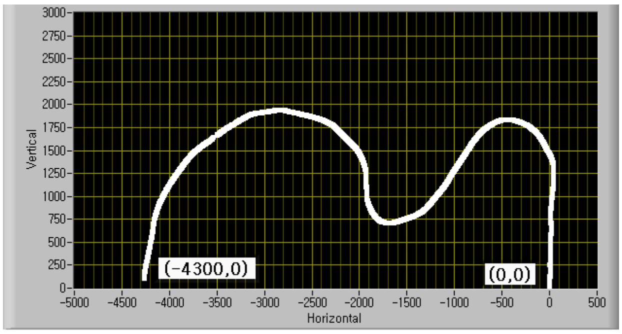


**Figure 12.** Experimental result of obstacle avoidance while navigating from the starting position (0, 0) to the target position (-4300, 0) in a cluttered environment

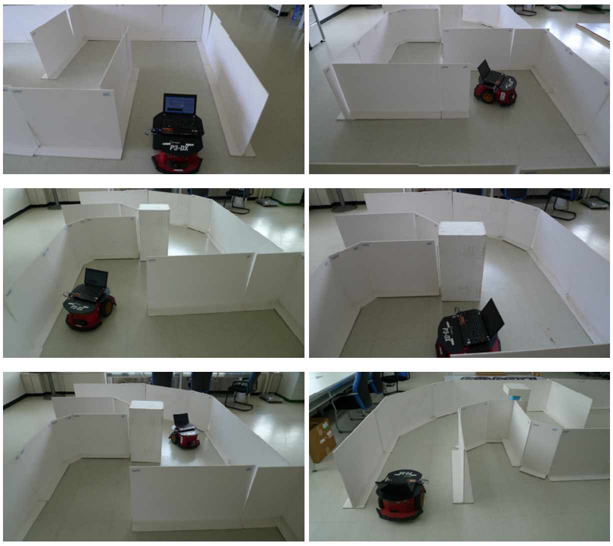


**Figure 13.** Experiment process in a cluttered environment for the result shown in Fig. 12

## 7. Conclusions

This work sought to implement a real-time mobile robot system that uses relatively less power. To perform faster operations in real-time systems, using faster processors may be enough for the controller. However, there are limitations when designing a low-power system with only high-performance processors. It is necessary to reduce the operation clock frequency to lower the power consumption because a high clock frequency elevates the power consumption of the system. All of these factors make it clear that a reduction of the operational complexity is a better answer for both faster operation and lower power consumption.

We utilized LUT(s) to eliminate the calculation of the membership degrees, which can be complicated and reiterative. Thus, we designed a full-LUT-based fuzzy controller, with which the average operation time is reduced to 20% of the time required by the fuzzy controller without LUT(s). However, the full-LUT design requires the buffers to contain all of the membership function values. To reduce the buffer size, we attempted to share the LUT for the membership functions. While the simulation results showed that the proposed LUT sharing method requires 50% more operation cycles on average compared to the full-LUT-based design, the proposed method saves 94.5% of the original LUT buffer size. The results show that the LUT sharing method can be more worthwhile for real-time mobile robots because it is approximately 3.35 times faster than the use of a fuzzy controller without LUT(s).

In conclusion, this work can provide advantages to real-time obstacle avoidance systems for use in mobile robots because the LUT-based fuzzy controller design enables faster obstacle avoidance while the LUT sharing method provides compactness of the controller by reducing most of LUT buffers.

## 8. Acknowledgments

This work was supported by the DGIST R&D Program of the Ministry of Education, Science, and Technology (MEST) of Korea (10-IT-03).

## 9. References

[1] Koren Y, Borenstein J, Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of IEEE Int. Conf. Robotics and Automation*, pp. 1398 – 1404, April 1991.

[2] Borenstein J, Koren Y, "The vector field histogram-fast obstacle avoidance for mobile robots," in *Proceedings of IEEE Int. Conf. Robotics and Automation*, pp. 278 – 288, June 1991.

[3] Thongchai S, Kawamura K, "Application of fuzzy control to a sonar based obstacle avoidance mobile robot," in *Proceedings of IEEE Int. Conf. Control Applications*, pp. 425 – 430, Sep. 2000.

[4] L.-X. Wang, "Adaptive Fuzzy Systems and Control: Design and Stability Analysis," *Prentice-Hall*, Englewood Cliffs, New Jersey, 1994.

[5] R. Palm, D. Driankov, and H. Hellendoorn, "Model Based Fuzzy Control: Fuzzy Gain Schedulers and Sliding Mode Fuzzy Controllers," *Springer-Verlag Berlin*, USA, 1997.

[6] K. M. Passino and S. Yurkovich, "Fuzzy Control," *Addison Wesley Longman*, Menlo Park, CA, 1998.

[7] E. Aguirre, A. Gonzalez, "Fuzzy behaviors for mobile robot navigation," *International Journal of Approximate Reasoning*, pp. 255-289, 2000.

[8] A. De Santis, B. Siciliano and L. Villani, "A unified fuzzy logic approach to trajectory planning and inverse kinematics for a fire fighting robot operating in tunnels," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 41-49, 2008.

[9] Indrani Kar and Laxmidhar Behera, Visual motor control of a 7 DOF robot manipulator using a fuzzy SOM network, *Intelligent Service Robotics*, vol. 3, no. 1, pp. 49-60, 2010.

[10] Mbede J B, Shugen Ma, Lei Zhang, Toure Y, Graefe V, "Robust neuro-fuzzy navigation of mobile manipulator among dynamic obstacles," in *Proceedings of IEEE Int. Conf. on Robotics and Automation*, pp.5051 – 5057, May 2004.

[11] John H. Lilly, "Evolution of a Negative-Rule Fuzzy Obstacle Avoidance Controller for an Autonomous Vehicle," *IEEE Trans. Fuzzy Systems*, vol.15, no.4, pp. 718 - 728, August 2007.

[12] H. Kiendl, "Design of advanced fuzzy systems," in *Proceedings of the TOOLMET'99 Symposium—Tool Environments and Development Methods for Intelligent Systems*, 1999.

[13] J. S. Branson, "A Trainable fuzzy system incorporating dynamically variable asymmetric spreads and negative rule defuzzification," in *Louisville, KY, Ph.D. dissertation Univ. of Louisville*, May 2000.

[14] http://msdn.microsoft.com/en-us/robotics.