VNU Journal of Science: Mathematics – Physics, Vol. 32, No. 3 (2016) 19-33

A Novel Behavior-based Navigation Architecture of Mobile Robot in Unknown Environments

Nguyen Thi Thanh Van*, Phung Manh Duong, Dang Anh Viet, Tran Quang Vinh

VNU University of Engineering and Technology, Hanoi, Vietnam

Received 08 September 2016
Revised 20 September 2016; Accepted 30 September 2016

Abstract: This study proposes behavior-based navigation architecture, named BBFM, for mobile robot in unknown environment with obstacles. The architecture is carried out in three steps: (i) analyzing the navigation problem to determine parameters of the architecture; (ii) designing the objective functions to relate input data with the desired output; and (iii) fusing the output of each objective function to generate the optimal control signal. We use fuzzy logic to design the objective functions and multi-objective optimization to find the Pareto optimal solution for the fusion. A number of simulations, comparisons, and experiments were conducted. The results show that the proposed architecture outperforms some popular behavior- based architectures in navigating the mobile robot in complex environments.

Keywords: Behavior-based navigation, fuzzy logic, multi-objective optimization, mobile robot.

1. Introduction

Navigation is fundamental for mobile robot applications. In order to complete any given task, the robot first needs to have capability to safely reach the target [1]. Navigation of mobile robots thus has been receiving much research attention. The exiting methods can be classified into two main categories: *hierarchical architectures* and *reactive* or *behavior-based architectures* [2]. The hierarchical architecture operates through sequent steps of sensing, planning and acting based on known model of the environment. This architecture is thus appropriate for static and structured environments. For unknown or unstructured environments, the behavior-based architecture is often used. This approach splits a complex navigation task into sub-tasks or behaviors. Each behavior has its own objective and executes independently. They are then combined in accordance to the state of environment to generate a global response. As the combination only uses the local data, the behavior-based architecture does not need to have a global map of the environment. The division into behaviors additionally enables the modularization and extendability of the architecture.

The main challenge with the behavior-based architecture is the combination of behaviors, called command fusion, to achieve the navigation objective. Several techniques have been proposed such as switching [3], motor schema [4], and decentralized information filter (DIF) [5]. However, the most

*Corresponding author. Tel.: 84-912720780
Email: vanntt@vnu.edu.vn

popular one is the fuzzy-based technique, which was practically used in recent mobile robot navigation systems [6-10]. In this technique, each behavior is presented by one fuzzy controller. The command fusion is then the fusion of output fuzzy sets of controllers and the final control signal is the value of defuzzification. This technique is simple in implementation and quite efficient in navigation. The command fusion, however, is not optimal due to limitation of defuzzification methods [11], [12]. Each method often results in a different value of defuzzification. These values, in some cases, even conflict with each other.

In order to deal with the optimization problem in command fusion, a technique based on multi-objective optimization theory, called MOASM, was proposed [13]. This technique represents each behavior by an objective function that relates input parameters such as mechanical structure, kinematic model and environment dynamics with the degree of achievement of the control objective. These functions are then combined using the multi-objective optimization to find an optimal solution which maximizes them. However, the main drawback of this technique is the lack of process for designing the objective functions. These functions may so complicated that preventing the technique to be deployed in practice.

In this study, we propose a behavior-based navigation architecture, called BBFM, which inherits advantages of fuzzy logic to design the objective functions and multiobjective optimization to fuse the behaviors. In BBFM, each behavior is represented by a reduced fuzzy controller which only contains the fuzzification and fuzzy inference processes. As the result, the output of each fuzzy controller will be a function of input variables whose value presents the achievement of behavior objective, or in other words, the objective function. These functions thus can be used as inputs for a multi objective optimization process to find the optimal control signal. A number of simulations, comparisons, and experiments have been carried out and the results confirmed the efficiency of the proposed architecture in navigating the mobile robot in complex and unknown environments.

The structure of paper includes six sections. Section II presents the BBMF architecture in general. Section III describes the implementation of BBFM for the case of differential drive wheeled mobile robot. Section IV simulates and compares the BBFM with two other popular architectures. The experimental results are presented in Section V. The paper finishes with discussions and conclusions in Section VI.

2. Behavior-Based Navigation and the BBFM Architecture

In this section, we present two popular fusion tech- niques. One uses fuzzy logic and the other uses multiple objectives optimization. Based on them, the BBFM architecture is proposed.

A. Behavior-based navigation using fuzzy logic

In behavior-based navigation using fuzzy logic, each behavior is implemented by a fuzzy controller. Each fuzzy controller includes three modules: fuzzification, inference engine and command fusion. The fuzzification describes data via linguistic values, for example the distance is near or far, without requiring the system model so that it is suitable for uncertainty characteristics of unknown environment. The fuzzy inference is executed by ”If...Then” rules similarly to the human’s inference. Finally, command fusion generates the overall control signal in one of two ways shown in Fig.1: defuzzicating first and then combining individual decisions; or combining individual decisions first and then defuzzicating.

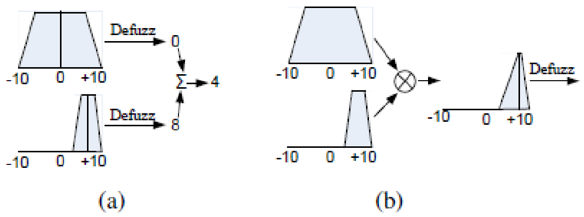


Fig. 1. Two approaches to command fusion: (a) Defuzzificating and then combining, (b) Combining and then defuzzificating.

Advantages of behavior-based navigation using fuzzy logic includes the ease in implementation and efficiency in navigation. However, the command fusion is not optimal. Fig.1 shows that the two ways of command fusion give different results. In addition, defuzzification methods such as centroid, mean of maximum or last of maximum produce different values. Consequently, the efficiency of navigation is not stable. In practice, we realize that the global control signal generated in some situations may even conflict with the output signal of certain behaviors.

B. Behavior-based navigation using multi-objective optimization

In behavior-based navigation using multi-objective optimization, each behavior is described by an objective function $O_k(\mathbf{y})$, where $\mathbf{y} = (y_1, y_2, ..., y_n) \in Y$ is the vector of control signal and Y is a set of possible actions, or control parameters. The optimal overall control signal is the solution of following equation:

$$\hat{\mathbf{y}} = \max[O_1(\mathbf{y}), O_2(\mathbf{y}), \ldots, O_N(\mathbf{y})]. \qquad (1)$$

According to the theory of multi-objective optimization, there does not exist the optimal solution, $\hat{\mathbf{y}}$, of Equation (1), but only the ”good enough” solution, $\mathbf{y}^*$, which is the best fit to all objectives O_i. This solution is called the Pareto optimal solution or non-dominated solution defining as follows: $\mathbf{y}^*$ is the Pareto optimal solution of Equation (1) if there does not exist any $\mathbf{y} \in Y$ such that $O_i(\mathbf{y}) > O_i(\mathbf{y}^*)$ at least one *i* and $O_j(\mathbf{y}) \geq O_j(\mathbf{y}^*)$ for all *j*. In other words, the Pareto optimal solution is the one in which there is not other solution that improves an objective without resulting in the deterioration of at least one other objective. Popular methods used to find the Pareto optimal solution includes the weighting, lexicographic and goal programming [14].

It is recognizable that the theory of multi-objective optimization provides a method to find the optimal solution for command fusion. However, it does not supply the method for defining objective functions. Without it, the deployment of this technique in practice is limited as the objective functions varies between systems and are often complex to manually define.

C. Behavior-based navigation architecture - BBFM

From the analyses, we realize that it is possible to inherit advantages of fuzzy logic and multi-objective optimization by using the output membership functions of fuzzy controllers as the objective functions for multiobjective optimization because each membership function maps the input space to the interval of [0, 1] representing the achievement of behavior objective. Fig. 2 shows the block diagram of BBFM. Each fuzzy controller is employed to build an objective function. The command fusion module then combines all objective functions using multi-objective optimization to generate the overall control signal. The deployment of BBFM is carried out in three steps: *task analysis, objective function design*, and *command fusion*. Details of each step are described as follows.

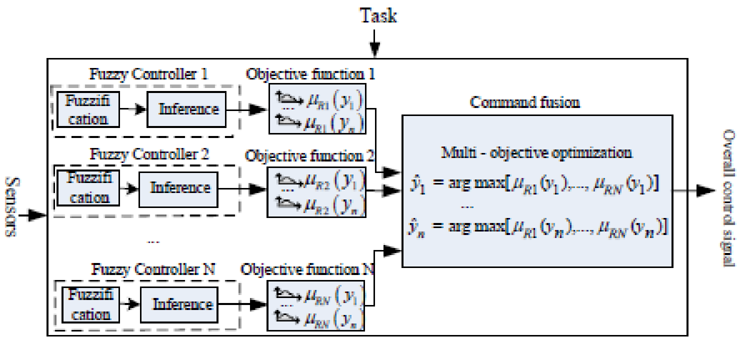


Fig. 2. The block diagram of BBFM architecture.

1) Task analysis

The purpose of task analysis is to determine main parameters for the BBFM architecture including the number of behaviors, their objectives, and the dimension of control signal. The number and objectives of behaviors are located based on the robot configuration, operating environment, and task assigned. The dimension of control signal depends on robot configuration and control method. Typically, outputs of all behavior need to have the same dimension to ensure the feasibility of command fusion: $\dim(\mathbf{y}_i) = \dim(\mathbf{y}_j)$.

2) Objective function design

Based on the parameters, a fuzzy controller is built for each behavior. It includes the fuzzification and fuzzy inference processes. The defuzzification is ignored. Consequently, the output of each fuzzy controller is a membership function which can be used as the objective function for the command fusion. Details of implementation are described as follows.

* *Fuzzification*

Fuzzification defines the input/output linguistic variables and their fuzzy sets. For each fuzzy controller, it is necessary to determine *m* input linguistic variables $\{x_1, x_2 \ldots x_m\}$ with the universe of discourse $\{X_1, X_{2\ldots} X_m\}$ and *n* output linguistic variables $\{y_{1,} y_{2\ldots,} y_n\}$ with the universe of discourse $\{Y_1, Y_2 \ldots Y_n\}$. Each input linguistic variable represents data from an input such as the distance measured by an ultrasonic sensor. Each output linguistic variable on the other hand represents a component of the control signal such as the tangent velocity. Values of a linguistic variable are determined by the fuzzy sets. Denoting a fuzzy set as A_{ij}, the input linguistic variable x_i and output linguistic variable y_i are then represented by:

$$\begin{aligned}
&x_1 = \{A_{11}, A_{12}, \ldots, A_{1a}\}\\
&x_2 = \{A_{21}, A_{22}, \ldots, A_{2a}\}\\
&\ldots\\
&x_m = \{A_{m1}, A_{m2}, \ldots, A_{ma}\}\\
&y_1 = \{B_{11}, B_{12}, \ldots, B_{1b}\}\\
&y_2 = \{B_{21}, B_{22}, \ldots, B_{2b}\}\\
&\ldots\\
&y_n = \{B_{n1}, B_{n2}, \ldots, B_{nb}\}
\end{aligned} \tag{2}$$

The membership functions are then represented by:

$$\begin{aligned}
&x_1 : (\mu_{A_{11}}(x_1), \mu_{A_{12}}(x_1), \ldots, \mu_{A_{1a}}(x_1)) \\
&x_2 : (\mu_{A_{21}}(x_2), \mu_{A_{22}}(x_2), \ldots, \mu_{A_{2a}}(x_2) \\
&\ldots \\
&x_m : (\mu_{A_{m1}}(x_m), \mu_{A_{m2}}(x_m), \ldots, \mu_{A_{ma}}(x_m)) \\
&y_1 : (\mu_{B_{11}}(y_1), \mu_{B_{12}}(y_1), \ldots, \mu_{B_{1b}}(y_1)) \\
&y_2 : (\mu_{B_{21}}(y_2), \mu_{B_{22}}(y_2), \ldots, \mu_{B_{2b}}(y_2)) \\
&\ldots \\
&y_n : (\mu_{B_{n1}}(y_n), \mu_{B_{n2}}(y_n), \ldots, \mu_{B_{nb}}(y_n))
\end{aligned} \tag{3}$$

where $\mu_{A_{ij}}$ is the membership function of input variables, $\mu_{B_{ij}}$ is the membership function of output variables.

** Fuzzy Inference*

Fuzzy inference is the process of building control rules and combining them to make output fuzzy sets. Each control rule, R_k, is of the form ”If...then...”, for instance:

If $x_1 = A_{11}$ and $x_2 = A_{21}$ and . . . $x_m = A_{m1}$ then $y_1 = B_{11}$ and $y_2 = B_{21}$ and . . . $y_n = B_{n1}$.

The result of above rule for each output control signal y_i is determined by:

$$\begin{aligned}
&\mu_{R_k}(y_1) = \min(H, \mu_{B_{11}}(y_1)) \\
&\mu_{R_k}(y_2) = \min(H, \mu_{B_{21}}(y_2)) \\
&\ldots \\
&\mu_{R_k}(y_n) = \min(H, \mu_{B_{n1}}(y_n)) \\
&H = \min\{\mu_{A_{11}}(x_1), \mu_{A_{21}}(x_2), \ldots, \mu_{A_{m1}}(x_m)\}
\end{aligned} \tag{4}$$

For M control rules, the implication R' of each y_i according to the max-min method is an output fuzzy set with the membership function defined by:

$$\mu_{R'}(y_i) = \max(\mu_{R_1}(y_i), \mu_{R_2}(y_i), \ldots, \mu_{R_M}(y_i)), O_N(\mathbf{y})] \tag{5}$$

The membership function (5) is the objective function of control signal y_i.

3) Command fusion

The command fusion generates a overall control signal by fusing outputs of all fuzzy controllers. Let N be the number of fuzzy controllers. Each component, y_i, of the control signal then has N objective functions determined by (5). According to multi-objective optimization theory, the Pareto optimal solution, $\hat{y}_i$, has to satisfy the following condition:

$$\overset{\boldsymbol{\mu}}{y}_i = \max[\mu_{R_1'}(y_i), \mu_{R_2'}(y_i), \ldots, \mu_{R_N'}(y_i)] \tag{6}$$

It can be found by using the Lexicographic method [14] as follows:

- Sorting all behaviors in descending order of importance, for example behavior 1, behavior 2, ..., behavior *N*.
- Sequentially solving equations Pi until an unique solution is obtained or all equations are solved:

$$\begin{aligned}
&P_1 : \max_{y_i \in Y_i} \mu_{R'_1}(y_i),\\
&P_2 : \max_{y_i \in Y_{i1}} \mu_{R'_2}(y_i),\\
&\ldots\\
&P_j : \max_{y_i \in Y_{i(j-1)}} \mu_{R'_j}(y_i),\\
&Y_{j(j-1)} = \{y_i \mid y_i \text{ is the solution of } P_{j-1}\},\\
&j = 2,\ldots,N+1
\end{aligned} \tag{7}$$

3. Implementation of BBFM for Differential Drive Wheeled Mobile Robot

This section presents the deployment of BBFM architecture for the differential drive wheeled mobile robot in unknown environments. Details of the steps of task analysis, objective function design and command fusion are described as follows.

A. Task analysis

This step determines parameters of the BBFM architecture based on the configuration of robot and task assigned.

1. Robot configuration

The robot is the type of differential drive wheeled mobile robot with nonholonomic constraints and parameters shown in Fig. 3, where (O_G, X_G, Y_G) is the global coordinate system; (O_R, X_R, Y_R) is the local coordinate system relative to the robot chassis; R is the wheel diameter; L is the distance between two wheels; (x, y, θ) represents the position and the direction of robot in the global coordinate system; ρ is the distance from the center of robot to the target; α is the angle between the axis of the robot's reference frame and the vector connecting the center of the axle of the wheels with the target position. The kinematic equation in discrete time domain of the robot is presented by [15]:

$$\begin{cases} x_{i+1} = x_i + u_i T_s \cos\theta_i \\ y_{i+1} = y_i + u_i T_s \sin\theta_i \\ \theta_{i+1} = \theta_i + \omega_i T_s. \end{cases} \tag{8}$$

where T_s is the sampling period, u_i and ω_i are respectively the tangential and angular velocity at sampling time i.

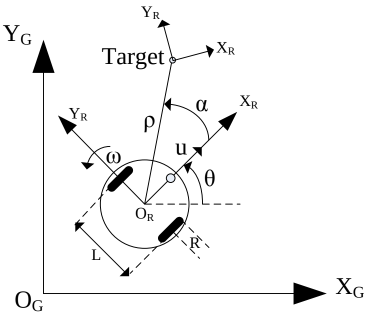


Fig. 3. Configuration of the differential drive wheeled mobile robot.

The robot is equipped with 8 ultrasonic sensors for obstacle detection. Each sensor has the measuring range from 0 m to 4 m and the scanning range of 15^0. They are arranged in front of the robot as shown in Fig.4 to cover the range of 160^0. In order to reduce the complexity of building fuzzy rules, all sensors are divided into three groups of Right (sensor 1, 2, 3), Front (sensor 4, 5) and Left (sensor 6, 7, 8). The measuring value of each group is the minimum value of all sensors in that group:

$$\begin{cases} d_{right} = min(d_1, d_2, d_3) \\ d_{front} = min(d_4, d_5) \\ d_{left} = min(d_6, d_7, d_8), \end{cases} \tag{9}$$

where d_i is the distance from sensor i to obstacle.

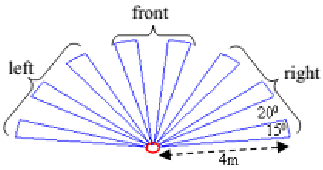


Fig. 4. Arrangement of utrasonic sensors on the robot.

2) Task assigned and parameters of BBFM architecture

The mission of the robot is to navigate in an unknown environment from an initial position to a desired target without colliding with obstacle. To complete this task, the controller uses the BBFM architecture with two behaviors: obstacle avoidance, and goal reaching. Each behavior is implemented by one fuzzy controller as shown in Fig. 5. Inputs include data of ultrasonic sensors measuring the distances from robot to obstacles and data of optical encoders measuring the pose of robot. Outputs are the tangential and angular velocities of robot: $\mathbf{y} = (u, \omega)$. The universes of discourse of outputs are set by limit velocities of robot: $u \in \mathrm{U} = [\mathrm{u}_{\min}, \mathrm{u}_{\max}], \omega \in \mathrm{W} = [\omega_{\min}, \omega_{\max}]$.

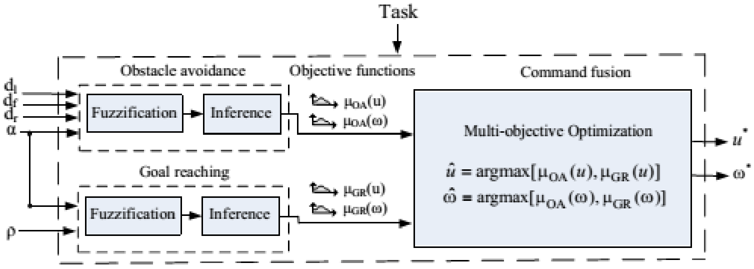


Fig. 5. The BBFM architecture designed for differential drive wheeled mobile robot.

B. Objective function design

Based on the parameters, we design a fuzzy controller for each behavior whose output is the desired objective function.

1) Obstacle avoidance controller

The obstacle avoidance controller includes four input variables and two output variables. Three input variables, d_{right}, d_{front}, *and* d_{left}, represent the far or near distance from robot to obstacle in right, front and left directions, respectively. Their crisp values are determined by Equation (9). The linguistic values contain Near (N), Medium (M), and Far (F):

$$d_{right} = d_{front} = d_{left} = \{N, M, F\} \tag{10}$$

The fourth input variable α is the deflection angle between robot and target defined by:

$$\alpha = \arctan(y_d - y, x_d - x) - \theta, \alpha \in [-\pi, \pi] \tag{11}$$

Its linguistic values contain Large Negative (LN), Negative (N), Zero (Z), Positive (P), Large Positive (LP):

$$\alpha = \{LN, N, Z, P, LP\} \tag{12}$$

Two output variables are u and ω. The linguistic values of u contain Small (S), Medium (M), and Large (L). The linguistic values of ω contain Large Negative (LNo), Negative (No), Zero (Zo), Positive (Po), and Large Positive (LPo):

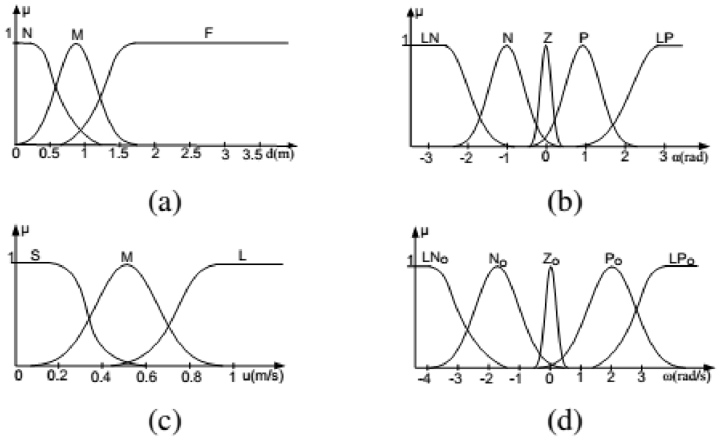
$$u = \{S, M, L\} \tag{13}$$

$$\omega = \{LNo, No, Zo, Po, LPo\} \tag{14}$$

The membership functions of input and output variables, as shown in Fig. 6, have the Gaussian and Sigmoid shapes defined by following equations:

$$Gauss(x) = e^{\frac{-x(x-c)^2}{2\sigma^2}} \tag{15}$$

$$Sigmoid(x) = \frac{1}{1 + e^{-a(x-b)}} \tag{16}$$



(a) (b) (c) (d)

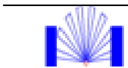
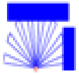
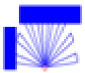
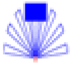
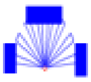
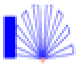
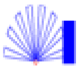
Fig. 6. Membership functions of input and output variables: (a) dleft, dfront, dright; (b) α; (c) u; (d) ω.

Table I presents 28 control rules defined for obstacle avoidance. Results of implication for u and ω according to the max-min method are given by:

$$\begin{aligned} \mu_{R_{OA}}(u) &= \max(\mu_{R_1}(u), \mu_{R_2}(u), \ldots, \mu_{R_{28}}(u)) \\ \mu_{R_{OA}}(\omega) &= \max(\mu_{R_1}(\omega), \mu_{R_2}(\omega), \ldots, \mu_{R_{28}}(\omega)) \end{aligned} \tag{17}$$

where $\mu_{R_k}(u)$ and $\mu_{R_k}(\omega)$ are results of kth rule defined in Equation (4).

Table 1. Rules denided for obstacle avoidance

Collisions	Rule	Input				Output	
		d_{left}	d_{front}	d_{right}	α	u	ω
	1	N	N	F		S	Po
	2	F	N	N		M	Po
	3	M	N	N		M	Po
	4	F	N	M		N	LPo
	5	N	N	F		M	LNo
	6	N	N	M		M	No
	7	F	N	F		M	LPo
	8	M	N	F		M	No
	9	F	N	M		M	Po
	10	N	M	N		M	Po
	11	N	N	N		S	Po
	12	M	N	M		S	Po
	13	N	M	M		M	No
	14	N	M	F		M	No
	15	N	F	M		M	No
	16	N	F	F	LN	S	LNo
	17	N	F	F	N	S	No
	18	N	F	F	Z	L	Zo
	19	N	N	N	LP	L	Zo
	20	N	F	F	P	L	Zo
	21	M	M	N		M	Po
	22	F	M	N		M	Po
	23	M	F	N		M	Po
	24	F	F	N	LN	L	Zo
	25	F	F	N	N	L	Zo
	26	F	F	N	Z	L	Zo
	27	F	F	N	LP	S	LPo
	28	F	F	N	P	S	LPo

2) Goal reaching controller

This behavior controls the robot to reach the target as fast as possible. For this task, it continuously adjusts the robot direction to match the goal direction while drives the robot at the fastest possible speed. Inputs of this controller include the distance, ρ, from the current position of the robot to the target and the deflection angle, α, between robot and target. Outputs are the tangential velocity u and angular velocity ω. Variables α, u, and ω have the same definition of linguistic values, universes of discourse, and membership functions as in the obstacle avoidance controller. Variable ρ is defined as:

$$\rho = \sqrt{(x_d - x)^2 + (y_d - y)^2} \tag{18}$$

The linguistic values of ρ are Near (N), Medium (M), and Far (F):

$$\rho = \{N, M, F\} \tag{19}$$

The universe of discourse of ρ is in the range of [0, 20]. The membership function of ρ has the shape of Gaussian and Sigmoid as shown in Fig. 7.

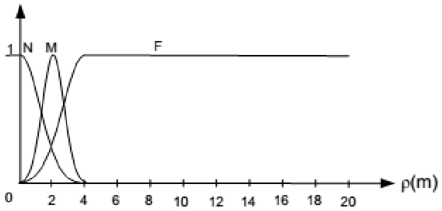


Fig. 7. The membership function of ρ.

The controller has 15 rules defined in Table II. Results of implication for u and ω according to the max-min method are given by Equation (20).

$$\mu_{R_{GR}}(u) = \max(\mu_{R_1}(u), \mu_{R_2}(u), \ldots, \mu_{R_{15}}(u))$$
$$\mu_{R_{GR}}(\omega) = \max(\mu_{R_1}(\omega), \mu_{R_2}(\omega), \ldots, \mu_{R_{15}}(\omega)) \tag{20}$$

Table 2. Rules defined for goal reaching

Rule	Input		Output	
	ρ	α	u	ω
1	N	Z	S	Zo
2	N	N	S	No
3	N	LN	S	LNo
4	N	P	S	Po
5	N	LP	S	LPo
6	M	Z	M	Zo
7	M	N	M	No
8	M	LN	M	LNo
9	M	P	M	Po
10	M	LP	M	LPo
11	F	Z	L	Zo
12	F	N	L	No
13	F	LN	L	LNo
14	F	P	L	Po
15	F	LP	L	LPo

C. Command fusion

Command fusion is implemented based on multiobjective optimization theory in which the objective functions are the output membership functions of (17) and (20). The optimal overall control signal $(\hat{u},\hat{\omega})$ is determined by:

$$\hat{u} = \max[\mu_{R_{OA}}(u), \mu_{R_{GR}}(u), \mu_{R_{DE}}(u)]$$
$$\hat{\omega} = \max[\mu_{R_{OA}}(\omega), \mu_{R_{GR}}(\omega), \mu_{R_{DE}}(\omega)] \tag{21}$$

The Lexicographic method is used to find the Pareto optimal solution of (21) as follows:

- Sorting all behaviors in descending order of importance: obstacle avoidance, and goal reaching.

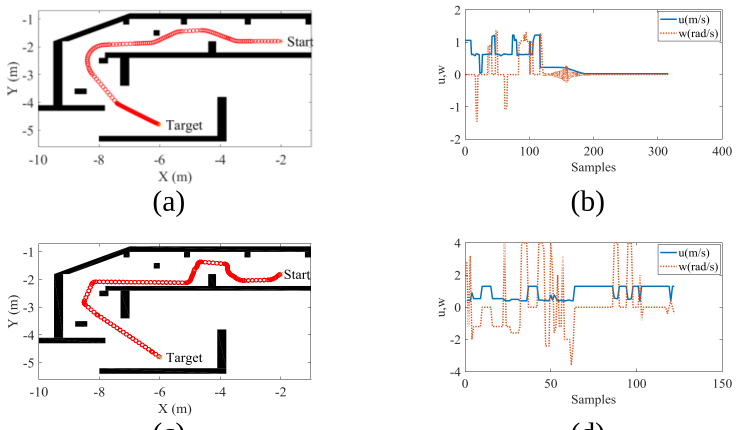
- Sequentially solving equations P_i by using discrete values of u and ω on set **U** and **W** until a unique solution is obtained or all equations are solved:

$$u^*:\begin{cases} P_1:\max\limits_{\mathrm{u\hat{I}U}}[\mu_{R_{OA}}(u)],\\ P_2:\max\limits_{u\in\mathbf{U_1}}[\mu_{R_{GR}}(u)],\\ \mathbf{U_1}=\{u\,|\,u\quad\text{solves}\quad \mathrm{P_1}\} \end{cases}$$

$$\omega^*:\begin{cases} P_1:\max\limits_{\omega\in\mathbf{W}}[\mu_{R_{OA}}(\omega)],\\ P_2:\max\limits_{\omega\in\mathbf{W_1}}[\mu_{R_{GR}}(\omega)],\\ \mathbf{W_1}=\{\omega\,|\,\omega\quad\text{solves}\quad \mathrm{P_1}\} \end{cases} \tag{22}$$

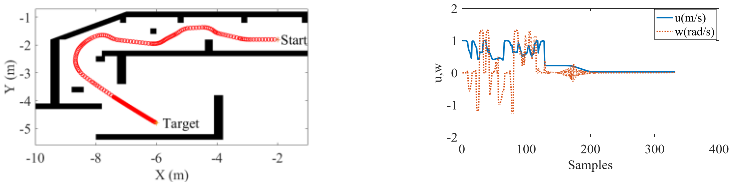
4. Simulations

Simulations have been implemented to evaluate the efficiency of BBFM compared to two other popular architectures including the MOASM [13] and CDB [7]. MOASM uses multi-objective optimization and CDB uses fuzzy logic. MOASM uses multi-objective optimization and is implemented with three behaviors: obstacle avoidance, maintaining target heading and moving fast forward. The objective functions of these behaviors are built based on the principle of Instantaneous Center of Curvature (ICC) of differential drive wheeled mobile robot. The overall control value is determined by using the Lexicographic method. The CDB uses fuzzy logic and is also implemented with three behaviors as in MOASM. However, the overall control value is determined by fuzzy-meta rules and deffuzification. In order to ensure the equality between architectures in comparison, the BBFM uses the obstacle avoidance and goal reaching controllers. All architectures are stimulated in Matlab with the same condition of operating environment and robot configuration. Parameters for simulations are set as follows: R = 0.085 m, L = 0.265 m, u € [0, 1.3] m/s, ω € [−4.3, 4.3] rad/s. The comparison results in three different cases are presented as follows.



(a) (b)

(c) (d)



(e) (f)

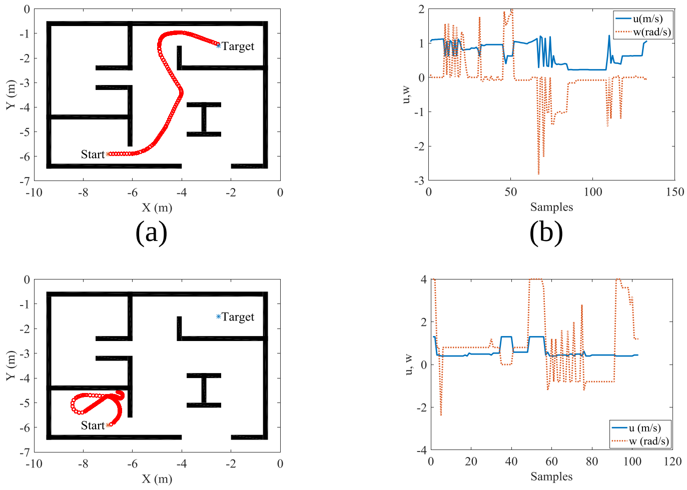
Fig. 8. The path and velocity responses of robot in three architectures in Case 2: (a) and (b): BBFM, c) and (d): MOASM, (e) and (f): CDB.

Table 3. Navigation results in Case 1

Parameters	BBFM	MOASM	CDB
Traveling path (m)	10.36	11.02	11.02
Time to reach to the target (s)	28.26	41.45	36.43
Error at the target (m)	0.05	0.2	0.05

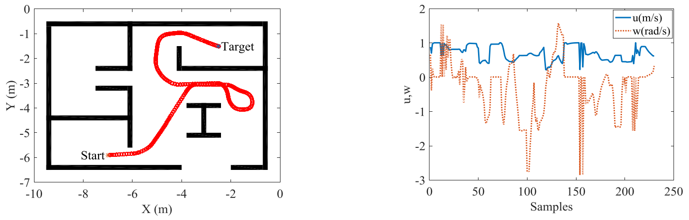
Case 1: The operating environment is chosen to be the same as in the original paper of MOASM [13]. The start position is (-2, -1.8, 180^o) and the target position is (-6, -4.8, 0^o). Fig. 8 shows the path of robot generated by three architectures: MOASM, BBFM, and CBD. Table III compares the traveling path, time to reach the target and error at the target. It shows that the BBFM is more effective than the remaining architectures in sense of smaller traveling path, faster time to reach the target, and smaller error at the target.

Case 2: The environment is chosen to be more like an office with obstacles which are walls and bulkheads. The start position is (-7, -6, 0^o) and the target is (-2.5, -1.5, 0^o). Fig. 9 and Table IV show the navigation results in which the BBFM controls the robot to reach the target with the shortest path and fastest time; the CDB requires longer path and time; and the MOASM does not complete the navigation task as the robot falls into a local minimum region.



(a) (b)

(c) (d)



(e) (f)

Fig. 9. The path and velocity responses of robot in three architectures in Case 2: (a) and (b): BBFM, c) and (d): MOASM, (e) and (f): CDB.

Table 4. Navigation results in Case 2

Parameters	BBFM	CDB
Traveling path (m)	9.35	15.66
Time to reach to the target (s)	12.09	24.45
Error at the target (m)	0.05	0.05

5. Experiments

In order to evaluate the operation of BBFM in real environments, we carried out experiment under different conditions. Details of setup and result are presented as follows.

A. Experimental Setup

The robot used in experiments is a Sputnik robot of DrRobot Inc [16] as shown in Fig. 10. It equips three ultrasonic sensors DUR5200 at left, front and right directions creating the scanning range from −60^o to 60^o. In order to open the scanning range to [−90^o, 90^o], we added two ultrasonic sensors SRF05 to the left and right sides of the robot, each employs a micro controller PIC12F1572 to synchronize data from SRF05 with the main board of Sputnik robot. The maximum tangential and angular velocities of robot are set to 0.5 m/s and 3.77 rad/s, respectively. The position of robot is determined via optical encoder sensors. The robot has a wireless module connecting it with a Wifi router (Fig. 10). The BBFM is written in Matlab and installed in a PC which communicates with the robot through the Wifi router. The BBFM receives data of sensors via the network, processes it, and sends the overall control command to the robot. Parameters of BBFM are set as follows: {d_{lelf}, d_{front}, d_{right}} € [0, 2.5] m, *u* € [0, 0.5] m/s, *ω* € [−3.7, 3.7] rad/s, T_s = 300 ms. The experimental environment is an indoor office with size of 4 m x 3 m and changeable obstacles.

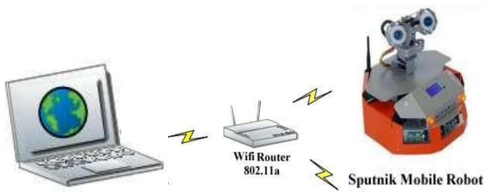
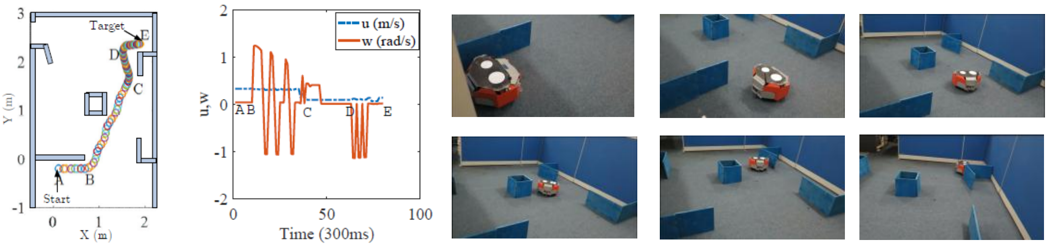


Fig. 10. The Sputnik robot and its configuration for communication with the control computer

B. Experimental Results

Fig. 11 presents the paths, velocity responses and photos of robot operation in lab environment with unknown obstacles. The robot starts at (0.1, -0.2, 0^o), then goes following the wall to B. At B, it turns left and avoids obstacle to C. Then the robot goes straight to D and adjusts its direction to avoid the bulkhead corners to reach the target E (1.8, 2.3, 0^o) as shown in Fig. 11(a). Fig. 11(b) shows the correspondence of linear and angular velocities of the robot with those movements. The velocity average of 0.157 m/s determined by travelled distance (3.77 m) per elapsed time (24 s) implies that the operation of robot is stable and suitable for the indoor environment. Some of photos of real operating are shown in Fig. 11(c).



(a) (b) (c)

Fig. 11. The results of navigating operations: (a) Path, (b) Velocity responses, (c) Photos.

6. Conclusions

In this paper, we have proposed a new behaviorbased navigation architecture, BBFM, for navigating the mobile robot in unknown environments. It inherits advantages of fuzzy logic to design objective functions and advantages of multi-objective optimization to fuse control signals. The architecture is simple to implement via three steps of problem analysis, objective function design, and command fusion. It is also flexible to extend by adding/removing behaviors to adapt to different navigation tasks. Simulations, comparisons, and experiments were conducted and the results show that the proposed architecture is high efficiency in term of accuracy, traveling path, and time response for the task of navigating in unknown environments with unpredictable obstacles

Acknowledgements

This work has been supported by VNU, University of Engineering and Technology under project number CN16.03.

References

[1] S. Roland and N. I. R, “Introduction to autonomous mobile robots,” The MIT Press Cambridge, vol. Massachusetts London, England, 2004.

[2] S. B. M. N. D. Nakhaeinia, S. H. Tang and O. Motlagh, “A review of control architectures for autonomous navigation of mobile robot,” International Journal of the Physical Sciences, vol. 6, no. 2, pp. 169–174, Jan. 2011.

[3] Dorigo.M and Comombetti.M, “Robot shaping: an experiment in behavior engineering,” MIT Press/Bradford Books, 1997.

[4] R. Arkin, “Motor-schema based mobile robot navigation,” Int. J. Robot, vol. 8, no. 4, 1989.

[5] M. S.-F. Eduardo Freire, Teodiano Bastos-Filho and R. Carelli, “A new mobile robot control approach via fusion of control signals,” IEEE transactions on system, mam and cybernetics, vol. 34, no. 1, Feb. 2004.

[6] A. Adriansyah and S. H. M. Amini, “Genetic fuzzy system in behavior based mobile robot,” Politeknik Elektronika Negeri Surabaya ITS, 2004.

[7] A. S. A. Yahmedi and M. A. Fatmi, “Fuzzy logic based navigation of mobile robot,” Recent Advances in Mobile Robotics, Dec. 2011.

[8] Q. T. Hongwei Mo and L. Meng, “Behavior - based fuzzy control for mobile robot navigation,” Mathematical Problems in Engineering, no. Article ID 561451, 2013.

[9] E. E.-P. T. AlYahmedi, A.S, “Behavior based control of a robotic based navigation aid for the blind,” Control and Applications Conference.

[10] R. H.-H. M. M. A. Mohammed Faisal, Khalid Al-Mutib and E. Mattar, “Behavior based mobile for mobile robots navigation and obstacle avoidance,” International Journal of computers and communications, vol. 8.

[11] E.H.Ruspini, “Fuzzy logic in the flakey robot,” In Proc of the Int.Conf on Fuzzy Logic and Neural Networks, Iizuka, Japan,1990.

[12] Saffiotti, “The uses of fuzzy logic in autonomous robot navigation,” Soft Computing, Springer Verlag, 1997.

[13] P. Pirjanian, “Multiple objective behavior-based control,” Robotics and Autonomous Systems, Elsevier, no. 31, 2000.

[14] A. Ben Tal, “Characterization of pareto and lexicograhpic optimal solutions,” Lecture Note in Economics and Mathematical Systems, Spinger, Berlin, 1990.

[15] L. T. . I. S. . G. K. car, “Ekf-based localization of a wheeled mobile robot in structured environments,” J Intell Robot Syst, no. 62.

[16] D. R. Manual, http://www.drrobot.com/products.