

## DESIGN OF GOAL-SEEKING BEHAVIOR-BASED MOBILE ROBOT USING PARTICLE SWARM FUZZY CONTROLLER

<sup>a</sup>Andi Adriansyah, <sup>b</sup>Badaruddin, <sup>c</sup>Eko Ihsanto

<sup>a,b,c</sup>Electrical Engineering Departement, Faculty of Engineering, Universitas Mercu Buana  
Jl. Meruya Selatan, Kembangan, Jakarta Barat, 11650, Indonesia  
Email: <sup>a</sup>andi@mercubuana.ac.id

### *Abstract*

*Behavior-based control architecture has successfully demonstrated their competence in mobile robot development. There is a key issue in behavior-based mobile robot namely the behavior design problems. Fuzzy logic system characteristics are suitable to address the problems. However, there are difficulties encountered when setting fuzzy parameters manually. Therefore, most of the works in the field generate certain interest for the study of fuzzy systems with added learning capabilities. This paper presents the development of fuzzy behavior-based control architecture using Particle Swarm Optimization (PSO). Then, goal-seeking behaviors based on Particle Swarm Fuzzy Controller (PSFC) are developed using the modified PSO with two stages of the PSFC process. A new nonlinear function of modulated inertia weight adaptation with time, named as Sigmoid Decreasing Inertia Weight (SDIW), is designed for improving the performance of PSO. Several simulations and experiments with MagellanPro mobile robot have been performed to analyze the performance of the algorithm. The promising results have proved that the proposed control architecture for mobile robot has better capability to accomplish useful task in real office-like environment.*

*Keywords: behavior-based robot; fuzzy logic; PSO; PSFC.*

## INTRODUCTION

Designing a mobile robot is a challenging task. Generally, the mobile robot should face complex environment, perceive imprecise sensor and act with imperfect actuator in fast response. Behavior-based control architecture is an alternative approach suitable to address these problems [1][2][3][4] [5]. The architecture is able to act with fast real-time response, provides for higher-level deliberation and has demonstrated its reliable performance in standard robotic activities. However, a kind of soft computing is needed to perform two key issues in behavior-based systems, such as generating optimal individual behavior and coordinating multiple behaviors.

Currently, several methods that hybrid fuzzy system with evolutionary algorithms has been proposed in behavior-based mobile robot, such as Genetic Algorithm [6][7], Genetic Programming [8] to overcome the behavior-based issues. However, the existing evolutionary algorithms used have several drawbacks, such as not easy to implement and computationally expensive [9], require much process should be completed and parameters should be adjusted, have slow convergence ability to find near-optimum solution, and dependent heuristically to genetic operators [10].

Fortunately, Kennedy and Eberhart introduced the Particle Swarm Optimization (PSO) in 1995 [11] and [12]. PSO is one of evolutionary computation technique to find the optimal solution by simulating such social behavior of groups such as fish schooling or bird flocking. There are several advantages of the PSO as compared to other evolutionary computation methods. The PSO is easy to implement and is computationally inexpensive since its memory and CPU speed requirements are low [9] and [10]. Additionally, the PSO requires only a few process should be completed and parameters to be adjusted. In another side, the PSO has quick convergence ability to find optimum or near-optimum solution. Generally, PSO has proved to be an efficient method for numerous general optimization problems, and in some cases it does not suffer from the problems encountered by other evolutionary computation [11].

This paper addresses the problems of developing control architecture of mobile robot

with behavior-based system, especially in goal-seeking behavior. The problem solving is related to the specification of mobile robot tasks, the development of mobile robot behaviors, the interpretation of the environment and the validation of the final system. This paper uses and develops soft computing, making extensive use of Fuzzy Logic and Particle Swarm Optimization (PSO) named as Particle Swarm Fuzzy Controller (PSFC). The use of PSO is to tune fuzzy membership function and to learn fuzzy rule base for goal-seeking behavior. This fuzzy tuning and learning is performed to accomplish the best behavior-based system

## GOAL SEEKING BEHAVIOR MODEL

Goal seeking behavior steers and moves the robot to the right direction and reach the goal effectively. The mobile robot movement towards the goal is according to the distance and angle between the current position of the mobile robot and the goal position [13] and [14].

In this work, MagellanPro mobile robot is used for verification and performance analysis of the proposed algorithm. The MagellanPro is a circular mobile robot from iRobot, Real World Interface (RWI), the acknowledged industry leader in the exciting field of cutting-edge mobile robotic. The dimension of the robot is as follows:  $D = 40.64$  cm,  $H = 25.4$  cm,  $r = 5.7$  cm,  $W = 36$  cm and  $M = 18.2$  kg, where  $D$  is diameter,  $H$  is height;  $r$  is the radius of wheels,  $W$  is distance between two wheels, and  $M$  is weight, respectively. Figure 1 shows the physical structure of MagellanPro mobile robot.



Figure 1. MagellanPro Mobile Robot

Figure 2 illustrated a model of MagellanPro mobile robot for simulation exercises for the

proposed algorithm. The mobile robot is located on a two dimensional Cartesian workspace, in which a global coordinate  $\{X, O, Y\}$  is defined. The robot has three degrees of parameter position that are represented by a posture  $p_c = (x_c, y_c, \theta_c)$ , where  $(x_c, y_c)$  indicate the spatial position of the robot guide point in the global coordinate system and  $\theta_c$  is the heading angle of the robot counter-clockwise from the x-axis.

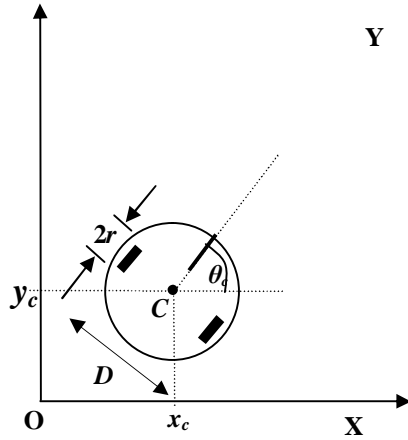


Figure 2. Model of MagellanPro mobile robot

The mathematical model for the robot movement can be obtained with differentially steered drive system or known as differential drive system [15]. Based on this system, the robot can move to different positions and orientations as a function of time. The derivatives of  $x$ ,  $y$  and  $\theta$  can be obtained as shown in Equation (1), Equation (2), and Equation (3).

$$\frac{dx}{dt} = v_c \cos \theta_c \quad (1)$$

$$\frac{dy}{dt} = v_c \sin \theta_c \quad (2)$$

$$\frac{d\theta}{dt} = \omega_c \quad (3)$$

where  $\omega_c$  is the angular velocity of the robot and where  $v_c$  is the linear velocity of the robot. By applying the current position of the robot,  $p_c = (x_c, y_c, \theta_c)$ , the next position of the robot is

shown in Equation (4), Equation (5), and Equation (6).

$$x_{c+1} = x_c + v_c \cos \theta_c * \Delta t \quad (4)$$

$$y_{c+1} = y_c + v_c \sin \theta_c * \Delta t \quad (5)$$

$$\theta_{c+1} = \theta_c + \omega_c * \Delta t \quad (6)$$

Then, as assuming the value of  $\Delta t$  is a unit time step, the next position of the robot,  $p_{c+1} = (x_{c+1}, y_{c+1}, \theta_{c+1})$ , in simple form is in Equation (7), Equation (8), and Equation (9).

$$x_{c+1} = x_c + v_c \cos \theta_c \quad (7)$$

$$y_{c+1} = y_c + v_c \sin \theta_c \quad (8)$$

$$\theta_{c+1} = \theta_c + \omega_c \quad (9)$$

In maintaining a course to a goal location, or searching, an effective strategy called as aiming navigation [16] is used. The mobile robot aiming at a goal has to orient its body axis such that the goal is in front of it. The goal must be associated with some salient cue. In this work, the goal is prior specified by the human user. Furthermore, by means of odometry strategy, both the direction and the distance to the goal are acquired. Finally, the goal can be approached from various directions, as illustrated in Figure 3. This technique is simple, fast, and has no cumulative error reroute to the goal.

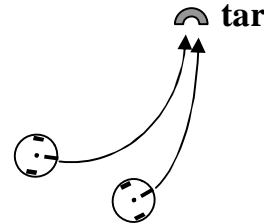


Figure 3. Aiming navigation

In the direction of moving to a goal point, mobile robots need to know its relative position. By some modification, integrating and applying the initial position of the robot as  $x(0) = x_0$ ,  $y(0) = y_0$ , and  $\theta(0) = \theta_0$  yields:

where  $v_r$  and  $v_l$  are the right and the left wheel velocities along the ground, respectively, and the position of the robot at time  $t$  at the coordinate is  $x(t)$ ,  $y(t)$ ,  $\theta(t)$ .

Based on the robot position and heading, the relative position to the goal point is calculated. The relative positions are known as target distance ( $d$ ) and target angle ( $\delta$ ) as seen in Equation (13) and Equation (14).

$$d = \sqrt{(x(t) - x_g)^2 + (y(t) - y_g)^2} \quad (13)$$

$$\delta = \arctan\left(\frac{(y(t) - y_g)}{(x(t) - x_g)}\right) - \theta(t) \quad (14)$$

where  $p_g = (x_g, y_g, \theta_g)$  is the goal position. The target distance ( $d$ ) and target angle ( $\delta$ ) are used as the inputs for goal seeking behavior as in

$$x(t) = x_0 + \frac{W(v_r + v_l)}{2(v_r - v_l)} \left[ \sin\left(\frac{(v_r - v_l)t}{W} + \theta_0\right) - \sin\theta_0 \right] \quad (10)$$

$$y(t) = y_0 - \frac{W(v_r + v_l)}{2(v_r - v_l)} \left[ \cos\left(\frac{(v_r - v_l)t}{W} + \theta_0\right) - \cos\theta_0 \right] \quad (11)$$

$$\theta(t) = \theta_0 + \frac{(v_r - v_l)t}{W} \quad (12)$$

Equation (10), Equation (11), and Equation (12).

Figure 4 illustrates the relative position between the robot ( $p(t)$ ) and the goal point ( $p_g$ ).

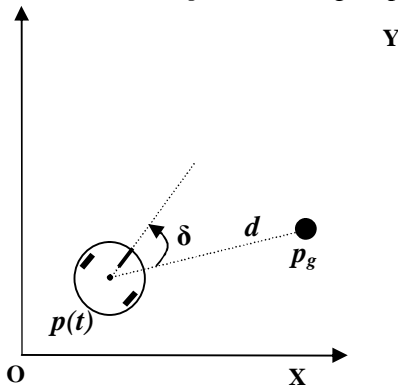


Figure 4. Relative positions between the robot and the goal point

## FUZZY GOAL SEEKING BEHAVIOR STRUCTURE

FLC structure based on Mamdani technique is used in this system. There are two inputs required, named as target distance ( $d$ ) and target angle ( $\delta$ ). These inputs are obtained by

calculation of the relative position between the current position of robot by means of odometer and the goal position, as stated in Equation (13) and Equation (14). Also, there are two output resulted, named as linear velocity,  $v$ , and angular velocity,  $\omega$ .

Trapezoid and triangular shape are used as input membership functions and output membership functions, for fuzzification and defuzzification process, respectively. The relational function between input and control of a fuzzy behavior are described by means of fuzzy rule base. Each rule is concatenated as:

$$RB_i: \text{if } X_1 \text{ is } A_1 \text{ and } X_2 \text{ is } A_2 \text{ and } \dots \quad (15)$$

$$X_n \text{ is } A_n \text{ then } Y_1 \text{ is } B_1 \text{ and } Y_2 \text{ is } B_2$$

Then, the output is obtained by applying the fuzzy rule base inference and the centroid defuzzification scheme, as

$$y_o = \frac{\sum_{i \in l} \alpha_i \cdot C_{ox} D_{ox}}{\sum_{i \in l} \alpha_i \cdot D_{ox}} \quad (16)$$

where  $C_{ox}$  and  $D_{ox}$  are the parameters of center and width of output membership functions at rule  $i$ ,  $\alpha_i$  is the product of the degree of membership of each inputs at rule  $i$ , and  $l$  is the total number of rules fired. The selected techniques were chosen due to their linearity, computational simplicity, and easy to understand.

Every input of fuzzy has three linguistic terms, which are CLOSE, MEDIUM and FAR for distances and RIGHT, FORWARD and LEFT for angle, as depicted generally in Figure 5. Three linguistic terms is chosen on behalf of the minimal number for fuzzy system. The value of  $x_i$  and  $y_i$  are tuned automatically as described in the next sections.

In this work, linear velocity,  $v$ , and angular velocity,  $\omega$ , are applied as outputs of all fuzzy behavior modules. The linguistic terms used are LOW, MEDIUM and HIGH for linear velocity, and, RIGHT, FORWARD, and LEFT for angular velocity. The fixed membership functions of  $v$  and  $\omega$  is shown in Figure 6.

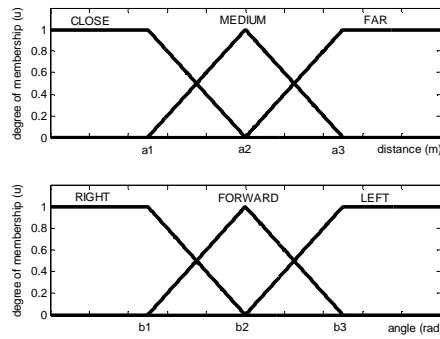


Figure 5. The membership functions of distances and angle

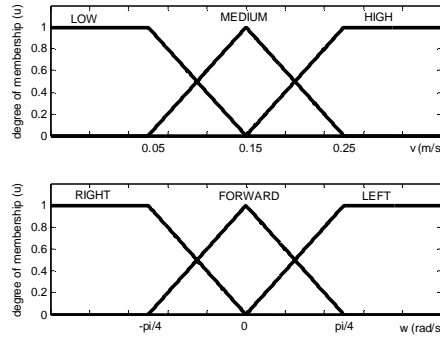


Figure 6. The membership function of linear velocity and angular velocity

## PARTICLE SWARM OPTIMIZATION

PSO is one of the artificial life or multiple particles' type techniques designed and developed by Kennedy and Eberhart in 1995 [11][12]. The method finds the optimal solution by simulating such social behavior of groups as fish schooling or bird flocking. That is, PSO is an optimization method that uses a principle of social behavior of a group.

The concept of PSO can be described as follows: each potential solution, called *particle*, knows its best value so far (*pbest*) and its position. Moreover, each particle knows the best value in the group (*gbest*) among the *pbest*. All of the best values are based on fitness function ( $F(.)$ ) for each problem to be solved. Each particle tries to modify its position using the current velocity and its position. The velocity of each particle can be calculated using Equation (17).

$$v_i^{k+1} = v_i^k + c_1 * rand(.) * (pbest - s_i^k) + c_2 * rand(.) * (gbest - s_i^k) \quad (17)$$

where  $v_i^k$ ,  $v_i^{k+1}$ , and  $s_i^k$ , are velocity vector, modified velocity and positioning vector of particle  $i$  at iteration  $k$ , respectively, *pbest* and *gbest* are best position found by particle  $i$  and best position found by particle group, and finally,  $c_j$  and  $w_j$  are weight coefficients for each term and inertia weight function for velocity of particle  $i$ , respectively.

Afterward, the current position of a particle is calculated by Equation (18):

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (18)$$

Updating process of particle is depicted in Figure 7.

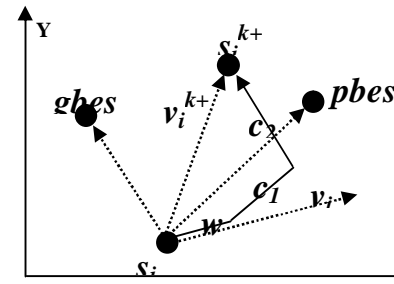


Figure 7. The velocity and position updates in PSO

An algorithm to find the best positioning vector of PSO using  $n$  particles can be summarized as follows:

1. Initial positioning vector  $S[n]$  and velocity vector  $V[n]$  are generated by using random values.
2. Velocity vector  $v_i^{k+1}$  of particle  $i$  is calculated by Equation (17).
3. Positioning vector  $s_i^{k+1}$  of particle  $i$  is updated by Equation (18).
4. If  $F(s_i^k)$  is better than the  $F(pbest_i)$ , the positioning vector  $s_i^k$  is set to *pbest*. If  $F(pbest_i)$  is better than  $F(gbest)$ , the positioning vector *gbest* is set to *pbest*.
5. If the iteration reaches to the pre-determined one, then stop. Otherwise, go to step 2.

To get better control exploration and exploitation of particles searched, the concept of inertia weight,  $w$ , is developed. Suitable selection of the inertia weight provides a balance between global and local searching. The inclusion of an inertia weight was first

reported in [17], where  $w$  is decreased linearly from about 0.9 to 0.4 during a run and has provided improved performance on a number of applications.

This paper proposed a sigmoid decreasing inertia weight (SDIW) to provide a better compromise of exploitation-exploration trade-off [18]. In SDIW, a large inertia weight is maintained in first part of PSO process to assure a global search. Afterwards, a small inertia weight is retained to facilitate a local search in final part of PSO process. There is very short inertia weight gradation between the large and small values. This method will provide a balance between global and local searching to give the PSO a superior performance.

## PARTICLE SWARM FUZZY CONTROLLER (PSFC)

Basically, PSFC is an FLC augmented by a tuning or learning process based on PSO. In PSFC, PSO is applied in order to search for an appropriate Knowledge Base (KB) of a fuzzy system for a particular problem and to ensure those parameter values are optimal with respect to the design criteria. The KB parameters constitute the optimization space, which is then transformed into suitable position on which the search process operates.

The PSO optimization process in FLC can be described by considering the functional block diagram as shown in Fig. 8. At the beginning of the process, the initial populations comprise a set of particles that are scattered all over the search space. The initial population may be randomly generated or may be partly supplied by the user. However, in this works, the populations are randomized initially.

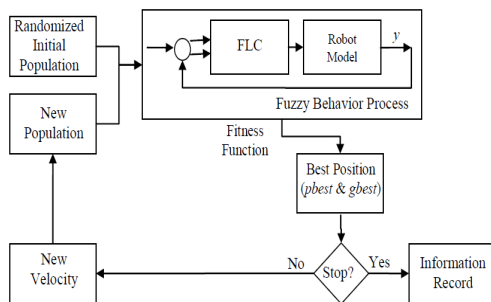


Figure 8. Block diagram of PSO operations in FLC process

Afterward, one particle is taken and decoded to the actual value of the fuzzy parameter. These sets of fuzzy controller parameters are then used to control the fuzzy behavior where it undergoes a series of tracking response of multistep reference set point. The use of a multistep reference signal is to excite the different states of the system, to enable the evaluation to cover a wider system operation range. Based on the various state of the control system, the performance of the controller is calculated by using a predefined fitness function. PSO is then used to tune the fuzzy controller parameters to minimize the fitness function. The assignment of the fitness function serves as a guidance to lead the search toward the optimal solution.

The KB consists of fuzzy membership functions (MF) and fuzzy rule base (RB). Consequently, there are several options to design PSFC, such as tuning membership functions, or learning for fuzzy rule base or tune/learn both of them in parallel. When tuning membership functions, an individual particle represents parameters of the membership function shapes at which fuzzy rule base is predefined in advance. However, when learning fuzzy rules base, the population represents all of fuzzy rules possibility using the membership functions that is assumed before. Therefore, it is proposed in this work to design PSFC in two stages of PSFC process, as depicted in Figure 9.

In the first stage, PSO starts to learn fuzzy rule base with predefined fuzzy membership function. In the next stage, PSO continues to tune fuzzy membership functions based on the results from the fuzzy rule base. By means of these two stages, ideal fuzzy parameter could be reached without human intervention.

## PSFC DESIGN

As the PSO deals with coded parameters, all parameters that need to be tuned or learned must be encoded into a finite length of string. The encoded strings are concatenated to form a complete particle. First, in order to learn fuzzy rule base, each rule is encoded into integer codes that are based on number in linguistic terms of output membership function. Consequently, there are '1', '2', and '3' for LOW, MEDIUM, and HIGH for linear

velocity, and RIGHT, FORWARD, and LEFT for angular velocity, respectively. The coded parameters for each behavior are arranged to form particles of the population, as shown in Equation (19).

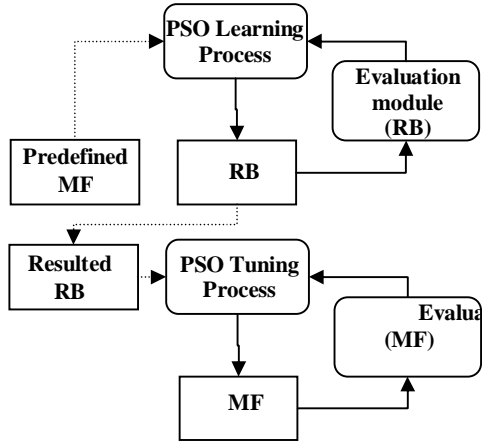


Figure 9. Two stages of PSFC process.

$$\begin{array}{l}
 \text{Particle} \quad |RB_{11}|, \dots, |RB_{1n}|, \dots, |RB_{21}|, \dots, |RB_{2n}|, \dots, |RB_{m1}|, \dots, |RB_{m2n}| \\
 \text{Parameter} \quad | \dots, (RB_{out1}), \dots, (RB_{out2}), \dots | \quad (19) \\
 \text{Result} \quad | \dots, \dots, \dots |
 \end{array}$$

where  $RB_{in}$  is the  $n$ -th rule base for  $i$ -th output.

Furthermore, for the purposed of tuning fuzzy membership functions the following equations were defined:

$$C_x = C_x + k_i \quad (20)$$

$$D_x = D_x + j_i \quad (21)$$

where  $k_i$  and  $j_i$  are adjustment coefficients,  $C_x$ , and  $D_x$  are set of centre and width of each fuzzy membership function, respectively. The adjustment coefficients take any real positive or negative value. Therefore,  $k_i$  makes each center of membership function move to the right or left and the membership functions shrinks or expands through  $j_i$ , as shown in Fig. 10. The shifting coding strategy will simplify searching computation, because there is no necessity to sort the value of membership functions in ascending manner.

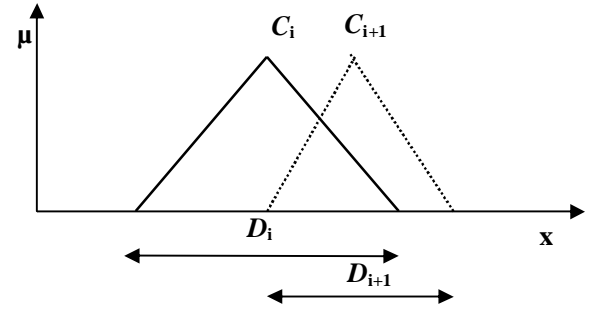


Figure 10. Principle in tuning of membership function

Then, the adjustment coefficients are encoded to form a complete particle, given as

$$\begin{array}{l}
 \text{Particle} \quad |k_{11}|, \dots, |k_{1i}|, \dots, |j_{11}|, \dots, |j_{1i}|, \dots, |k_{21}|, \dots, |k_{2i}|, \dots, |j_{21}|, \dots, |j_{2i}| \\
 \text{Parameter} \quad | \dots, (MF_1), \dots, (MF_2), \dots | \quad (22) \\
 \text{Result} \quad | \dots, \dots, \dots |
 \end{array}$$

where  $k_{mi}$  and  $j_{mi}$  are  $k$  and  $j$  at MF for the  $m$ -th input and the  $i$ -th linguistic term.

The PSO process starts with randomly generated initial populations. Then, all populations of particles are evaluated and associated based on fitness function to determine the  $pbest$  and  $gbest$ . Based on several initial investigations and trials and errors, the fitness functions for goal seeking can be obtained as stated in Equation (23).

$$f_{goal} = \sum_{i=0}^I \sum_{k=0}^K (100e_{\theta}^2(k) + e_d^2(k) + 100/v(k)) \quad (23)$$

where  $I$  is the total number of start position,  $K$  is the number of step simulation for each start position,  $e_{\theta}$  is the angle error,  $e_d$  is the distance error, and  $v(k)$  are the linear velocity at  $k$ , respectively. In this work, a Sigmoid Decreasing Inertia Weight (SDIW) is used to provide faster speed of convergence and better accuracy of optimized value. Consequently, PSFC would generate optimal and reliable goal-seeking behavior of the mobile robot.

## RESULTS AND DISCUSSION

Several experiments exercises have been performed. Some steps of experiments have been designed. Firstly, a PSFC optimization processes is conducted to find the optimized value of fuzzy parameters. Then, simulations

of the mobile robot based on the PSFC are analyzed to investigate the control behavior of PSFC. Results of fuzzy behavior that are obtained manually, obtained by GA, called as Genetic Fuzzy Controller (GFC) from previous works are used as comparison [19]. Finally, a real robot, MagellanPro, movements are tested to go to some location.

PSO and GA processes for goal-seeking behavior are shown in Figure 11, where, evolutions of the best fitness value against generation are illustrated. At the beginning of the run, the process tended to have more global search ability because of large inertia weight. It was shown that the fitness value over all generations is converging quickly. After that, the process tended to have more local search ability caused by the small value of inertial weight.

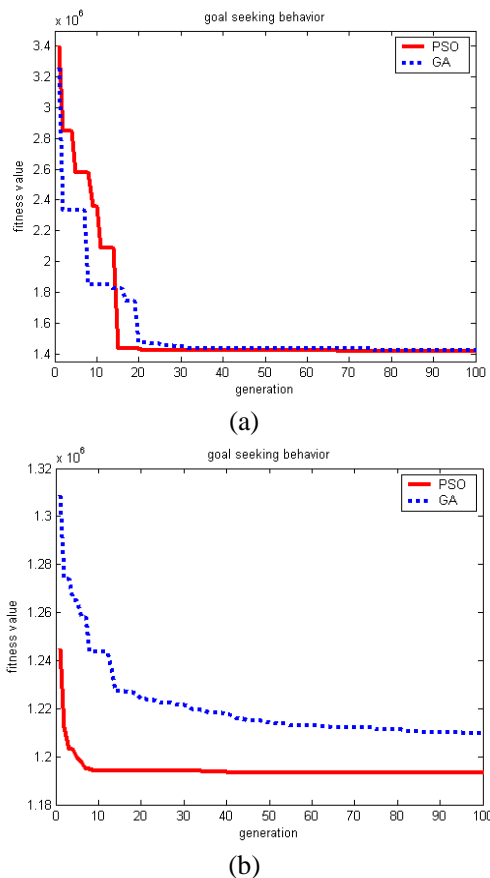


Figure 11. Comparison of PSO vs. GA process for goal seeking behavior ; (a) Rule base learning, (b) Membership function tuning

Hence, a goal was placed in certain position, (5, 8, 0) and depicted as a small black square.

Initially, robot are positioned at (2, 2,  $\pi/2$ ), (5, 2, 0), and (8, 2,  $\pi$ ). Simulations of mobile robot movements are depicted at Figure 12. It is noted that the PSFC was able to apply goal seeking behavior more effective than GA, in three cases.

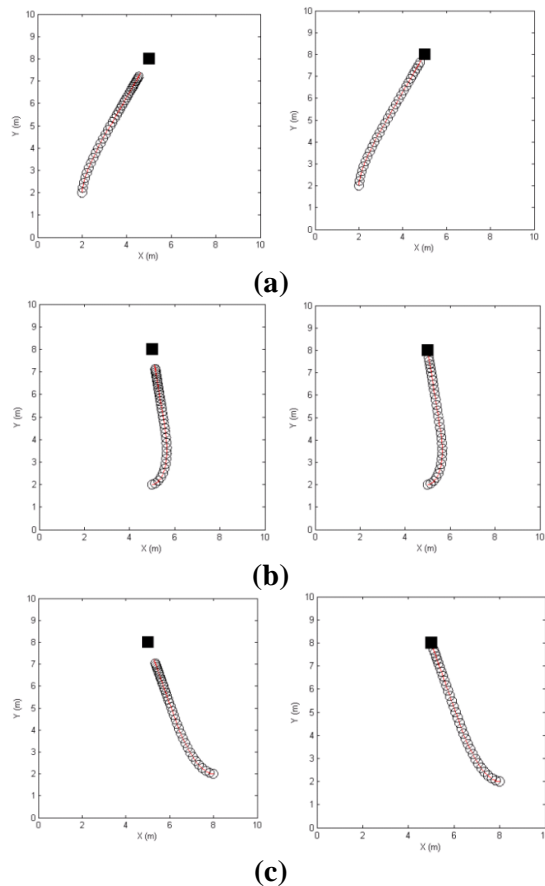


Figure 12. Simulation of mobile robot movements with GA(left) and PSO (right) comparison in differential initial positions

Finally, this experiment was performed to investigate the movement of the real mobile robot. The actual robot movement was depicted in Figure 13. The distance and angle of target from the current position as calculated from odometer inside the robot. The figure showed the performance of goal seeking behavior, where the mobile robot progressively reduced the distance and the angle between the target and the current position. The figure demonstrated the mobile robot was able to reach the target effectively.



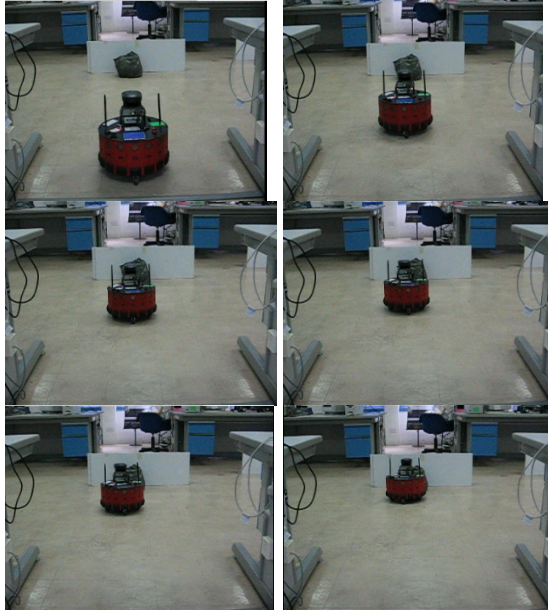


Figure 13. The actual robot movement

## CONCLUSION

Goal-seeking behavior-based control architecture has successfully demonstrated their competence in mobile robot development. Fuzzy Logic Systems appear to be very useful to develop the high reliable and effective behavior-based system. However, there are difficulties to set membership function and fuzzy rule base in Fuzzy System manually. This paper presents the development of fuzzy goal-seeking behavior-based control architecture using PSO for MagellanPro mobile robot. The work has been done in some tasks: behavioral designing of the mobile robot, designing new fuzzy behavior coordination, and finally, implementing the proposed algorithm in real environment.

Based on the experiment results, the mobile robot is able to deal with goal-seeking behaviors. Generally, it is noted that the proposed control architecture has the good ability to be applied in MagellanPro mobile.

## REFERENCES

- [1] W. Dongshu, Z. Yusheng, and S. Wenjie, "Behavior-Based Hierarchical Fuzzy Control for Mobile Robot Navigation in Dynamic Environment," *Chinese Control and Decision Control (CCDC 2011)*, China, May, 2011.
- [2] S. Parasuraman, V. Ganapathy, and B. Shirinzadeh, "Behaviour Based Mobile Robot Navigation Technique AI System: Experimental Investigation on Active Media Pioneer Robot," *IJUM Engineering Journal*, vol. 6, no.2, pp. 13-25, 2005.
- [3] Q.Y. Bao, S.M. Li, W.Y. Shang, and M.J. An, "A Fuzzy Behavior-based Architecture for Mobile Robot Navigation in Unknown Environments," in *Proceedings of International Conf. on Artificial Intelligence and Computational Intelligence (AICI 2009)*, Shanghai, November, 257-261, 2009.
- [4] H. Mo, Q. Tang, and L. Meng, (2013). "Behavior-based Fuzzy Control for Mobile Robot Navigation," *Mathematical Problems in Engineering*, pp. 1-10, 2013.
- [5] S. Khatoon and Ibraheem, "Autonomous Mobile Robot Navigation by Combining Local and Global Techniques," *International Journal of Computer Applications*, vol. 37, no. 3, pp. 1-10, 2012.
- [6] K. B. Sim, K.S. Byun, and D. W. Lee, "Design of Fuzzy Controller using Schema Coevolutionary Algorithm," *IEEE Trans. Of Fuzzy System*, vol. 12, no. 4, pp. 565-570, 2004.
- [7] E.A. Merchan-Cruz, A. S. Moris, "Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace," *IEEE Trans. On Robotics*, vol. 22, no. 4, pp. 613-624, 2006.
- [8] E.W. Tunstel, M.A.A. de Oliveira, and S. Berman. "Fuzzy Behavior Hierarchies for Multi Robot Control," *International Journal of Intelligent Systems. Special*

- Issue: Hierarchical Fuzzy Systems*, vol. 17, no. 5, pp. 449-470, 2002.
- [9] R. Hassan, B. Cohanin, and O. de Weck, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," *American Institute of Aeronautics and Astronautics*, pp. 1-13, 2004.
- [10] K.O Jones, "Comparison of Genetic Algorithm and Particle Swarm Optimization," *Proceeding of International Conference on Computer Systems and Technologies*. Vol. IIIA, pp. 1-6, 2005.
- [11] R.C. Eberhart and J. A Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan. Piscataway, N. J.: IEEE Service Center, pp. 39-43, 1995.
- [12] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," in *Proceeding of IEEE International Conference on Neural Networks*. IV. Perth, Australia. Piscataway, N. J.: IEEE Service Center, pp. 1942-1948, 1995.
- [13] V. R. Jisha and D. Ghose, "Goal seeking for robots in unknown environments," *International Conf. on Intelligent Robots and Systems (IROS 2010)*, Taipei, October, pp. 4692-4697, 2010.
- [14] L. Cherroun, and M. Boumehraz, "Designing of Goal Seeking and Obstacle Avoidance Behaviors for Mobile Robot using Fuzzy Techniques," *Journal Automation and System Engineering*, vol. 6, no.4, pp.164-171, 2012.
- [15] G. Dudek and M. Jenkin., *Computational Principles of Mobile Robotics*. 1<sup>st</sup> ed. Cambridge, MA: Cambridge University Press, 2000.
- [16] M.O. Franz and H.A. Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*. Vol. 30, pp. 133-153, 2000.
- [17] Y. Shi and R.C. Eberhart, "Empirical Study of Particle Swarm Optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Washington, DC. 1945-1950, 1999.
- [18] A. Adriansyah and S.H.M. Amin, "Analytical and Empirical Study of Particle Swarm Optimization with a Sigmoid Decreasing Inertia Weight," in *Proceedings of 1<sup>st</sup> Regional Postgraduate Conference on Engineering and Science (RPCES 2006)*, pp. 247-252, Malaysia, 26-27 July, 2006.
- [19] A. Adriansyah and S.H.M. Amin, "Knowledge Base Tuning using Genetic Algorithm for Fuzzy Behavior-based Autonomous Mobile Robot," in *Proceedings of 9<sup>th</sup> International Conference on Mechatronics Technology (ICMT 2005)*. Malaysia, December 5-8, 2005.