

## RESEARCH ARTICLE

# Intelligent Optimization of Adaptive Dynamic Window Approach for Mobile Robot Motion Control Using Fuzzy Logic

OMER ALI ABUBAKR<sup>1</sup>, MOHAMMAD A. JARADAT<sup>1,2</sup>,  
AND MAMOUN F. ABDEL-HAFEZ<sup>1,2</sup>, (Senior Member, IEEE)

<sup>1</sup>Mechatronics Engineering Graduate Program, American University of Sharjah, Sharjah, United Arab Emirates

<sup>2</sup>Department of Mechanical Engineering, American University of Sharjah, Sharjah, United Arab Emirates

Corresponding authors: Omer Ali Abubakr (b00071318@aus.edu.ae) and Mohammad A. Jaradat (mjaradat@aus.edu)

This work was supported in part by the Open Access Program from the American University of Sharjah.

**ABSTRACT** The paper presents an adaptive dynamic window approach (DWA) for mobile robot dynamic obstacles avoidance optimized utilizing a fuzzy logic controller. Most of the present work on autonomous navigation in dynamic environments does not take into account the dynamics of the obstacles. One of the methods used in research today for dynamic obstacles avoidance is the dynamic window approach. The (DWA) is a well-known navigation scheme. One problem facing the DWA is how to optimize the weights of its objective function to allow the robot to move towards the goal while avoiding collisions in all environments. The main contribution of this paper is to build an intelligent system that will be able to optimize the objective function weights of the dynamic window to make it more resilient to changes and moves as fast as possible towards the goal using fuzzy logic system. The proposed new adaptive controller was able to reduce the failure rate of the DWA from 20% to only five per cent in static environments, and maintain more than 60% success rate in dynamic environments with up to 25 obstacles/100 m<sup>2</sup>, on the other hand the basic algorithm was failing to less than 50% with 15 obstacles/100 m<sup>2</sup>.

**INDEX TERMS** Collision avoidance, dynamic window, fuzzy logic controller (FLC), mobile robots.

## I. INTRODUCTION

Autonomous mobile robot trajectory planning algorithms can be divided into two sections, path planning and obstacles avoidance algorithms. The path planning algorithms are used to generate a suitable path between two points from the robot starting position to the goal point depending on a model or a map of the environment, while on the other hand, the obstacle avoidance algorithms are concerned with avoiding collision with obstacles that may exist on the path calculated by the path planning algorithm. These obstacles can either be static or dynamic. Static obstacles can be considered directly on the path planning algorithm to generate a collision free path to the goal. However, dynamic obstacles locations are changing with time which make them difficult to consider

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

on the path planning algorithm. Re-planning of the path can be computationally expensive and time consuming. Hence, obstacles avoidance algorithms are generated to allow the robot to maintain its overall path while avoiding collisions with these obstacles which is based on motion commands rather than path planning.

Several types of algorithms were developed for obstacles avoidance such as the vector field histogram (VFH) [1], the lane curvature method (LCM) [2], bug algorithms, and Dynamic Window Approach (DWA) [3]. The latest showed superior performance compared to others [4], [5]. Intelligent algorithms were also deployed to develop obstacles avoidance solutions such as in [6], hybrid fuzzy potential field was proposed for autonomous mobile robot dynamic motion planning in dynamic environments. A parallel elite genetic algorithm was utilized for path planning for autonomous robot navigation in [7]. A hybrid learning approach was

presented in [8] for obstacle avoidance using a neuro-fuzzy controller based on supervised learning, a reinforcement learning method based on the fuzzy actor-critic learning algorithm was used to tune the algorithm after being implemented on the hardware to adapt to new environments without human intervention [8]. A behavior based intelligent controller was proposed in [9], [10] for mobile robot navigation. A fuzzy logic expert system was used to select among different behaviors based on behavior coordination module. The proposed approach provides superiority compared to conventional potential field approach. The reinforcement learning and deep reinforcement learning based approaches were also deployed to deal with the varying characteristic of dynamic environment for mobile robot navigation [11], [12]. This paper focuses on enhancing the dynamic window approach [3] that is used for dynamic obstacles avoidance by proposing an intelligent adaptive DWA to deal with the varying characteristic of dynamic environment. The dynamic window approach is an obstacles avoidance method that takes into account the kinematic and dynamic constraints of a mobile robot. Kinematic constraints are taken into account by directly searching the velocity space of a mobile robot. The search space is the set of tuples  $(v, \omega)$  of the translational velocity  $v$  and rotational velocities  $\omega$  that are achievable by the robot [13]. This search space is reduced to the admissible velocities allowing the robot to stop safely. Due to the limited accelerations of the motors, a further restriction is imposed on the velocities. The robot only considers velocities that can be reached within the next time interval. These velocities form the dynamic window which is centered on the current velocities of the robot in the velocity space.

Several enhancements were proposed on the original DWA algorithm. One of the methods to improve the performance of the DWA is the global dynamic window approach presented by Brock and Khatib in 1999 [13]. This method extends the original DWA by incorporating a simple and efficient motion planning algorithm without prior knowledge of the environment. H. Berti et al., used the Lyapunov stability criteria on the DWA [14] as a control strategy to guarantee and characterize goal arrival. The method uses an ideal control law that drives the robot to the goal guaranteeing a global convergence. A navigation function is used to drive the robot to the goal by avoiding collisions and considering the robot dynamic constraints.

In 2011, Chih-Chung Chou et al. introduced a new enhancement on the dynamic window approach called DWA\* [15]. Their method realizes the environmental information as interval configuration for fast processing. A candidate velocity is derived by analyzing the intervals to find navigable areas in the environment. For each candidate velocity, the new robot position is computed as a new node and saved in a search tree. This process is repeated until the goal is expanded or the tree depth reaches a certain value. The deepest node is then determined as a temporal goal.

Xiuyun LI et al., introduced an improved dynamic window approach in 2017 [16]. The algorithm is based on obstacle

sensing. The analysis of traverse feasibility is conducted by using the scanned data. The robot is modelled as rectangular and the scanning area is divided into 6 regions.

In 2015, Zhang Hong et al. suggested using fuzzy logic to change the weight parameters of the objective function for the dynamic window approach [17]. Their proposal to change the objective function weights is based on the distribution of obstacles around the robot in three different areas, target position, and target orientation. In 2018, O.A. Abubakr et al. presented a reduced design for the work of Zhang Hong et al. [18]. In their model they removed both the target orientation and target position from the weight optimization fuzzy controller. They introduced a new fuzzy controller with simplified rules set that enhances the performance of the fuzzy controller.

In the last decade, several attempts were made to address the problems of the dynamic window approach and to optimize its performance. P. Saranrittichai et al., published a paper on robust local obstacle avoidance [19]. In 2014, Lima et al. used an image based dynamic window approach for navigation in urban environments [20]. A modified DWA was proposed in [21] to limit the robot speed in crowded environment with many obstacles using fuzzy logic rules. In [22] the DWA was integrated with a feedback linearization controller for autonomous vehicle navigation. The DWA was utilized in [23] to provide advanced driving assist systems to control the linear and angular velocities of the vehicle to assist obstacle avoidance. Another driving assist system [24] utilized an Image-based DWA to implement human vehicle cooperative navigation system. In [25] the DWA and follow the gap method were used to implement real time obstacle avoidance navigation system for hovercraft. In [26] the DWA is utilized to implement a hybrid path planning approach for unmanned surface vehicle (USV) combined with A\* global path planner algorithm.

In this paper, we propose an intelligent fuzzy controller to optimize the weights of the objective function for the dynamic window approach. The fuzzy logic system exploits its ability to provide a robust system to deal with uncertainty and imprecision to solve the robot navigation problem in dynamic environment. Fuzzy logic system presents the weights of the dynamic window approach objective function by a flexible set of If- Then rules. These rules characterize the nonlinear dynamic behavior of the weight which reflects the expert system knowledge to adapt the weight. Thus the proposed system will be able to adapt the dynamic window approach weights based on the received complete or partial information from the surrounding robot environment [4], [6]. The proposed fuzzy controller depends on the distribution of obstacles in five areas around the robot. The fuzzy controller has a set of 32 rules that tune the three weight parameters of DWA based on the distance to the nearest obstacle in each of the five areas. The proposed controller is validated in both simulation and experiment using a Qbot open-architecture ground Quanser robot built on a Kobuki differential mobile platform [27], [28], [29]. The mobile robot platform is supported

with real-time based controller under MATLAB/Simulink using QUARC software from Quanser [30]. The results are compared with both the basic dynamic window approach with static weight parameters and another fuzzy controller in [17], [18].

Sections II and III describe the problem we intend to solve and the fundamentals of the DWA. The proposed intelligent solution is described in section IV. Section V evaluates the proposed approach and compares it to existing solutions in both simulation and experimental works. Finally, the concluding remarks are presented in section VI.

## II. PROBLEM STATEMENT

The main idea of the dynamic window approach is to select from all velocity tuples only the ones that allow the robot to stop before hitting an obstacle with respect to the kinematic capabilities of the robot. These velocities are called the admissible velocities. Hence, the dynamic window contains only those velocities that can be achieved by the robot within its dynamic limitations. This results in a search space to find the next motion command that allows the robot to approach its goal while avoiding collision with the nearest obstacle. After determining the search space, a velocity is selected based on the criteria target heading, clearance, and velocity using the objective function  $G(v, \omega)$  as follows:

$$G(v, \omega) = \alpha.heading(v, \omega) + \beta.dist(v, \omega) + \gamma.velocity(v, \omega) \tag{1}$$

where:

*Heading* = the alignment of the robot with the goal.

*Dist.* = the distance to the nearest obstacle to the robot.

*Velocity* = the linear velocity of the robot.

Figure 1 shows the main distances and angles required for the objective function of the dynamic window approach. From equation (1) and figure 1, the heading angle is the angle between the robot heading angle and the goal point  $(g_x, g_y)$  denoted as  $\Theta$  and can be calculated as follows:

$$\Theta = \psi - \varphi \tag{2}$$

$$\psi = \tan^{-1} \frac{g_y}{g_x} \tag{3}$$

where:

$\Theta$  = goal heading.

$\psi$  = goal angle with respect to the global frame.

$\varphi$  = robot heading angle with respect to the global frame.

Then:

$$heading = 180^\circ - \Theta \tag{4}$$

The heading is maximized for the objective function and this is performed using equation (4). The second parameter in the objective function is the (dist) which is the distance from the robot center  $(R_x, R_y)$  to the nearest obstacle  $(d_x, d_y)$  indicated as  $d_{obs}$  in figure 1 and is calculated as follows:

$$d_{obs} = \sqrt{(d_x - R_x)^2 + (d_y - R_y)^2} \tag{5}$$

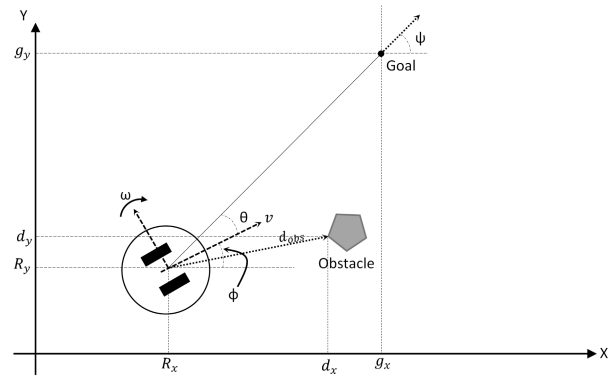


FIGURE 1. The main angles and distances to calculate the objective function.

The last component is the velocity and it is measured as the linear velocity of the robot on the axis  $v$  from the figure.

The three components of the objective function are necessary. For example, increasing the weights of only the clearance and velocity parameters of the objective function; the robot would be able to avoid collision with very high speed but with no incentive to move towards the goal location. This can be applied to any of the components which make all of them of equal effect over the whole approach.

The main objective of this paper is to find the parameters for these three components that give the optimal performance under any given condition by applying an intelligent algorithm that adjusts the weight effect of each component depending on the current location of the robot with respect to the nearest obstacle and the goal point. Also, depending on the current translational and rotational velocities of the robot.

## III. DYNAMIC WINDOW APPROACH (DWA)

The dynamic window approach was first proposed by D. Fox et al. [3]. The algorithm utilizes the sensory data from the robot with respect to the velocity as well as the obstacles around the robot to give the robot wheels the next suitable velocity command that will enable the robot to safely avoid collision and move towards the goal. Its main advantage is that it considers the robot kinematic constraints. The algorithm searches a well-chosen velocity space which is the combination of all possible set of tuples  $(v, \omega)$ . The concept of DWA is to get an optimal velocity command in the velocity space directly by maximizing the objective function.

Figure 2 demonstrates how the velocity space is calculated. There are three kinds of velocity spaces defined in DWA. The first velocity space is made up of all the possible velocities the robot can achieve  $V_s$ . Each velocity in the second velocity space ensures that the robot can stop before crashing into any obstacle if this velocity,  $V_a$ , is selected. The third velocity space contains the velocities that can be reached in a clock cycle, and it is also called the Dynamic Window  $V_d$ . The resultant velocity search space is given by  $V_r$ .

Based on the kinematics specification of the robot and the existence of obstacles in the environment the algorithm

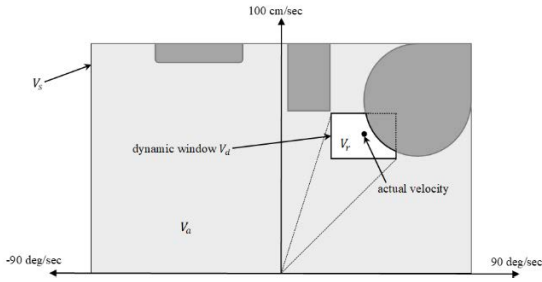


FIGURE 2. The dynamic window.

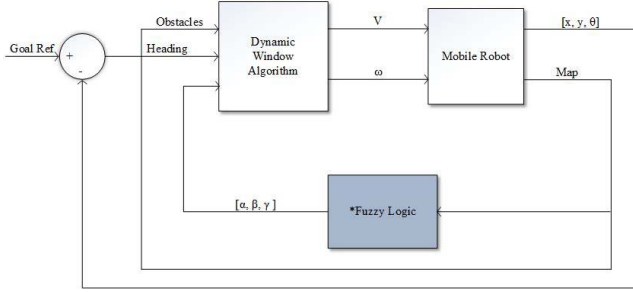


FIGURE 3. Block diagram of the fuzzy improved dynamic window approach (FIDWA).

calculates the dynamic window for the robot given by:

$$V_r = V_s \cap V_d \cap V_a \quad (6)$$

The second step is to pick the most appropriate velocity command. Firstly, each velocity in  $V_r$  is sampled to calculate its appropriateness with the objective function which is defined in equation 1 where  $\alpha, \beta$  and  $\gamma$  are constants and their summation is equal to 1.

From the equation, the heading is the angle between the robot heading angle and the goal location which is denoted by angle  $\Theta$  in figure 1.  $dist$  is the distance to the nearest obstacle and is denoted by  $d_{obs}$ . Velocity is the linear velocity of the robot which is used to force the robot to move towards the goal. After evaluating all the velocities using the objective function, the velocity command with the highest evaluation is selected as the next velocity command.

#### IV. FUZZY IMPROVED DYNAMIC WINDOW APPROACH

The fuzzy logic controller is applied to the DWA to optimize the weights of the objective function based on the existence of obstacles around the robot. Figure 3 shows the block diagram of the DWA system after adding the fuzzy controller. Figure 4 shows the different areas that are checked for obstacles. The area in front of the robot is divided into 5 regions with  $36^\circ$  for each region.

The five distance values from each area are used as inputs to the fuzzy controller. In this fuzzy controller Mamdani inference engine is used [31]. This inference engine uses (max–min) operator for composition, minimum operator for implication, maximum operator for aggregation and centroid for defuzzification. Firstly, input fuzzification is applied. Fuzzification is the process of converting the given crisp input

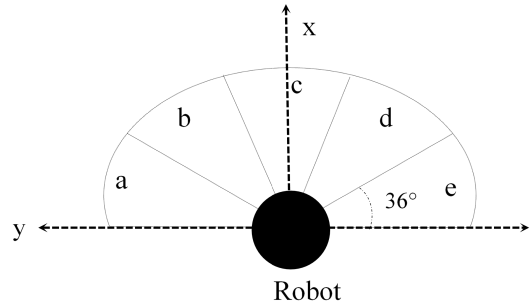


FIGURE 4. Obstacles areas around the robot.

values into their fuzzy values matching degree. Each input in the controller has two fuzzy terms: [near and far] with Gaussian bell membership for each term denoted as  $\mu_i$  where:

$$\mu_i(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (7)$$

Each of these parameters has a physical meaning.  $c$  determines the center of the corresponding membership function,  $a$  is the half width, and  $b$  with  $a$  controls the slopes at the crossover points. Figure 5 shows the fuzzy input for one of the areas. The fuzzy logic has three outputs, one for each of the weight elements  $[\alpha, \beta, \text{and } \gamma]$ . Every fuzzy output consists of five membership functions [very small, small, medium, high, and very high] with Gaussian bell shapes. Figure 6 shows the membership functions for one of the fuzzy outputs. All the fuzzy outputs are identical.

The implication process is the process of evaluating all the fuzzy rules with respect to the given fuzzy inputs [32]. For the developed fuzzy controller, a set of 32 rules are assigned for all the different cases and only a sample of the rules are shown here as follows:

- If (a is N) and (b is N) and (c is N) and (d is N) and (e is N) then ( $\alpha$  is M) ( $\beta$  is VH) ( $\gamma$  is S)
- If (a is N) and (b is N) and (c is F) and (d is N) and (e is N) then ( $\alpha$  is M) ( $\beta$  is M) ( $\gamma$  is M)
- If (a is N) and (b is F) and (c is N) and (d is N) and (e is F) then ( $\alpha$  is S) ( $\beta$  is VH) ( $\gamma$  is VH)
- If (a is N) and (b is N) and (c is N) and (d is N) and (e is N) then ( $\alpha$  is M) ( $\beta$  is VH) ( $\gamma$  is S)
- If (a is F) and (b is N) and (c is N) and (d is F) and (e is F) then ( $\alpha$  is S) ( $\beta$  is H) ( $\gamma$  is M)
- If (a is F) and (b is F) and (c is N) and (d is F) and (e is N) then ( $\alpha$  is S) ( $\beta$  is VH) ( $\gamma$  is M)
- If (a is F) and (b is F) and (c is F) and (d is F) and (e is F) then ( $\alpha$  is H) ( $\beta$  is M) ( $\gamma$  is M) ...

Tables 1 and 2 demonstrates the fuzzy associative memory of the intelligent system. Table 1 contains all the different possible output combinations of the objective function parameters while the input in area a is fixed to Near and table 2 when a is far.

The inference engine will use every rule and evaluate it with the inputs and set the output values based on the fuzzy

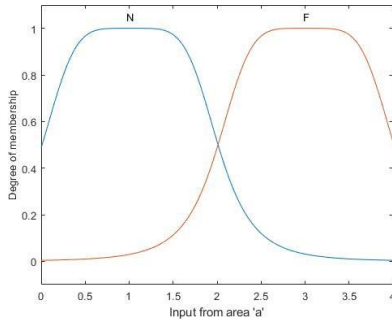


FIGURE 5. Fuzzy logic controller inputs membership functions.

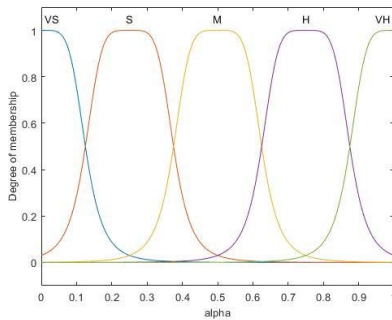


FIGURE 6. Fuzzy logic controller outputs membership functions.

TABLE 1. The fuzzy associative memory for Input a=N.

a=N	bc = NN	bc = NF	bc = FN	bc = FF
de=NN	$\alpha = M$ $\beta = VH$ $\gamma = S$	$\alpha = M$ $\beta = M$ $\gamma = M$	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$
de=NF	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$
de=FN	$\alpha = S$ $\beta = H$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$
de=FF	$\alpha = M$ $\beta = M$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$	$\alpha = S$ $\beta = VH$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$

outputs membership functions shown in figure 6 using the minimum operator ( $\wedge$ ) for rule input composition and rule output implication as in eq. (8) and (9), where \* can be  $\alpha$ ,  $\beta$ , or  $\gamma$ ). After the implication process, comes the aggregation of the consequents. In order to make a decision, all the rules implication must be combined together. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set using using the max operator ( $\cup$ ), as in eq. (10). In the last step, the defuzzification is implemented. The defuzzification is the process of converting the aggregated fuzzy set into a single number [33]. The method we are using for this step is the center of area centroid as in eq. (11).

$$\lambda_i = \mu_{a-i} \wedge \mu_{b-i} \wedge \mu_{c-i} \wedge \mu_{d-i} \wedge \mu_{e-i} \quad (8)$$

$$\mu_{*'-i} = \lambda_i \wedge \mu_{*-i} \quad (9)$$

TABLE 2. The fuzzy associative memory for Input a=F.

a=F	bc = NN	bc = NF	bc = FN	bc = FF
de = NN	$\alpha = S$ $\beta = H$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = H$	$\alpha = S$ $\beta = VH$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$
de = NF	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = H$	$\alpha = M$ $\beta = VH$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$
de = FN	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = H$	$\alpha = S$ $\beta = VH$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$
de = FF	$\alpha = S$ $\beta = H$ $\gamma = M$	$\alpha = H$ $\beta = M$ $\gamma = M$	$\alpha = S$ $\beta = VH$ $\gamma = S$	$\alpha = H$ $\beta = M$ $\gamma = M$

$$\mu_{*'} = \bigcup_{i=1}^R \mu_{*'-i} \quad (10)$$

where R is the number of rules.

$$x^* = \frac{\sum \mu_c(x) \cdot x}{\sum \mu_c(x)} \quad (11)$$

where x can be  $\alpha$ ,  $\beta$ , or  $\gamma$ .

## V. SIMULATION & EXPERIMENTAL RESULTS

### A. SIMULATION RESULTS

In this section, the proposed fuzzy improved dynamic window approach “FIDWA” is compared against the original dynamic window approach “DWA” [3] and the fuzzy dynamic window approach “FDWA” [17]. The results are compared in simulation with both static and dynamic obstacles. The performances of the algorithms are compared in terms of the path length required to reach the goal, the average linear velocity of the robot, and the time needed to reach the goal.

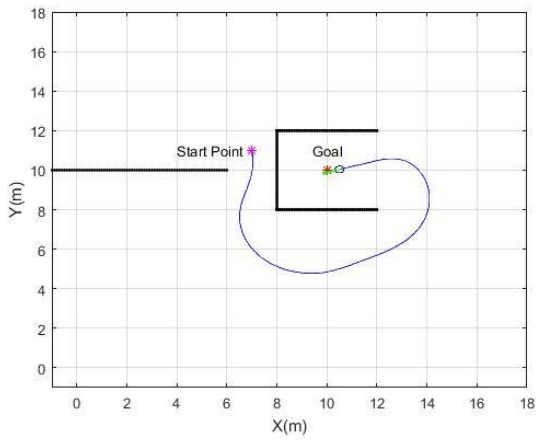
#### 1) SIMULATION VALIDATION IN A U-SHAPE ENVIRONMENT

In figure 7 the three algorithms are tested in an environment consisted of a U-shape wall where the robot is required to detect the wall and move around it to reach the goal point. From the figure, It can be noticed that FDWA was unable to move around the wall as the robot was driven directly towards the goal because of the small distance. The FIDWA gave the shortest possible path compared to the other approaches and reached the goal in approximately only 18 seconds.

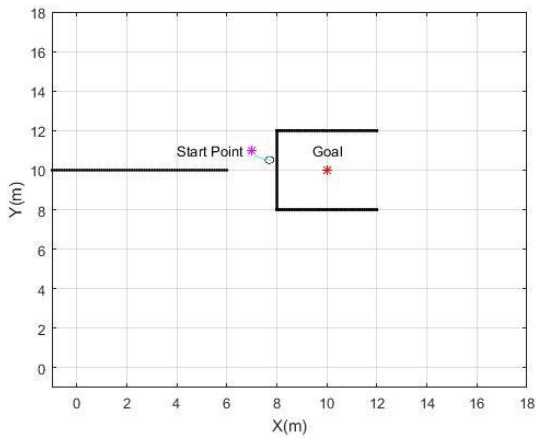
#### 2) SIMULATION VALIDATION IN A CORRIDOR ENVIRONMENT

In this test, the robot is required to enter a narrow corridor and move through to reach the goal point. In figure 8a the DWA algorithm was applied different combinations of weight parameters. The robot was unable to move through the narrow corridor in all combinations. In figure 8b the FDWA algorithm was applied in the corridor environment. The robot was able to enter the corridor but it failed to exit and was

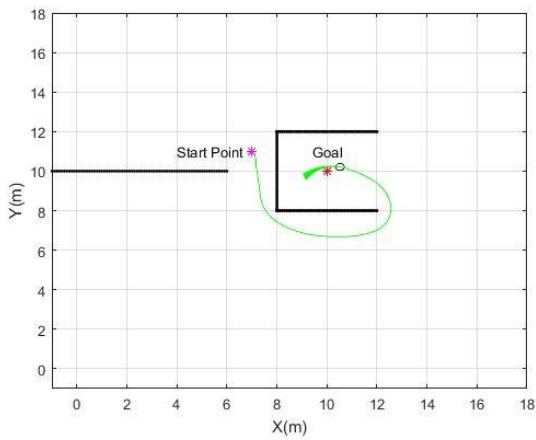




(a) DWA



(b) FDWA



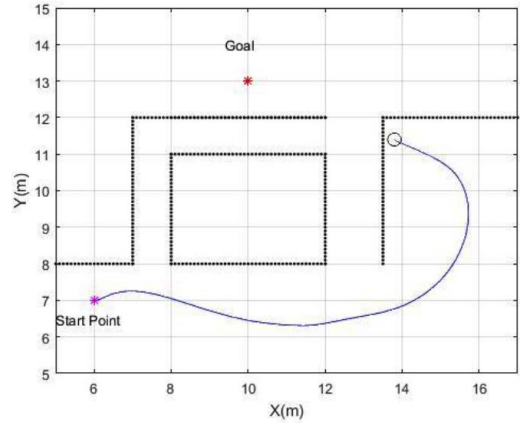
(c) FIDWA

FIGURE 7. U-shape wall testing scenario.

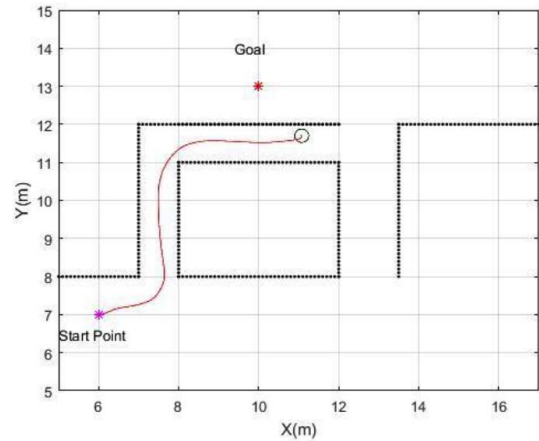
stuck inside. In figure 8c the FIDWA was tested in the same simulation environment. The robot was able to move in the narrow corridor and follow it until it exits the corridor and then move towards the goal point.

### 3) SIMULATION VALIDATION IN A STATIC ENVIRONMENT

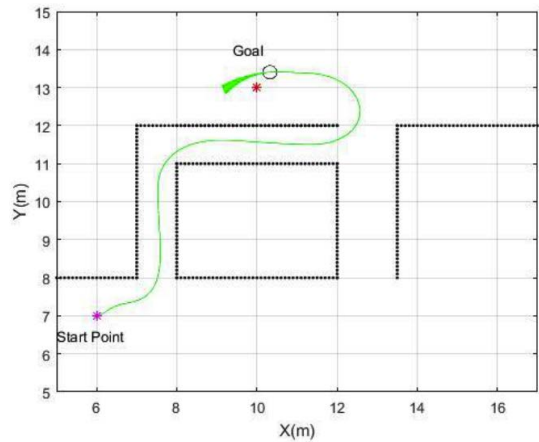
The performance of the three algorithms is tested with increasing number of static obstacles. Every algorithm was tested several times with specific number of static point



(a) DWA



(b) FDWA



(c) FIDWA

FIGURE 8. Corridor environment testing scenario.

obstacles and the success rate was recorded. Then the number of obstacles was increased and the process is repeated again. Figure 9 demonstrates the performance index of the three algorithms with increasing the number of obstacles from 10 to 30 obstacles. From the figure, the FIDWA is always having the best performance in all cases with a success rate of 80% or more for up to approximately 20 obstacles. The performance of the FIDWA drops to less than 50% success rate for only 30 obstacles or more between the robot and the goal point.

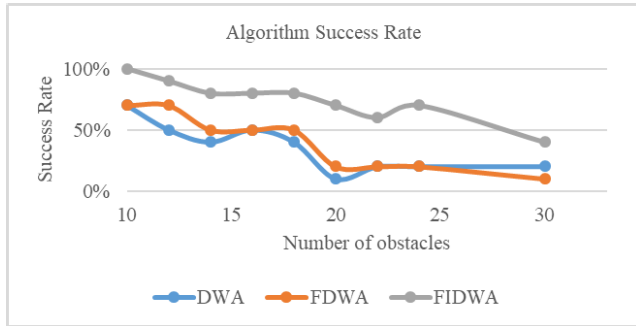


FIGURE 9. Algorithms success rate with static random point obstacles.

TABLE 3. Algorithms comparison for dynamic structured environment.

	Path length (m)	Average velocity (m/s)	Time(s)
DWA	24.4	0.64	40.1
FDWA	20.9	0.57	41
FIDWA	19.9	0.77	26.1

4) SIMULATION VALIDATION IN A DYNAMIC ENVIRONMENT

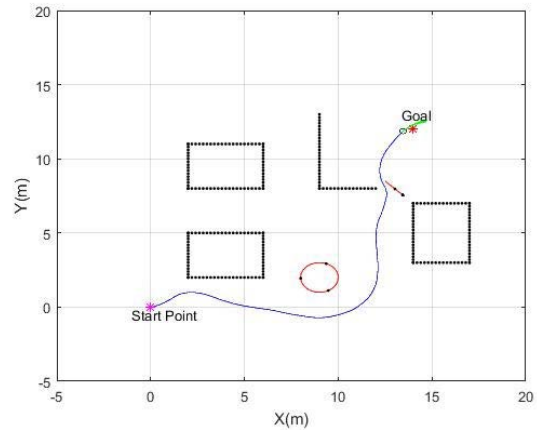
In the next test, the algorithms are tested in a dynamic environment as demonstrated in figure 10. The environment consists of several static obstacles and six dynamic point obstacles moving in a circle and a line. In this case all the algorithms were able to reach the goal point. Table 3 compares the results of the three algorithm. From the table, the shortest path was achieved using the FIDWA with the highest average linear velocity. Moreover, the FIDWA required the least amount of time to reach the goal point with 26 seconds.

Lastly, the failure rate of each algorithm is tested with moving obstacles. In this test a fixed number of randomly moving obstacle in the same area was simulated and the three algorithms were tested 40 times within the same environment to reach the goal point and the results were recorded. Figure 11 shows the results of this test. From the figure, it can be observed that the failure rate of the dynamic window approach to achieve its task was reduced considerably by using the proposed FIDWA. Moreover, the time required was also reduced while achieving a higher average velocity. Nevertheless the path length was on average the same between the FDWA and the FIDWA. Hence, we can say that the main contribution of the FIWDA is reducing the failure rate of the dynamic window approach to less than 5%, and reducing the average time required to reach the goal point by approximately 16% compared to the other controllers.

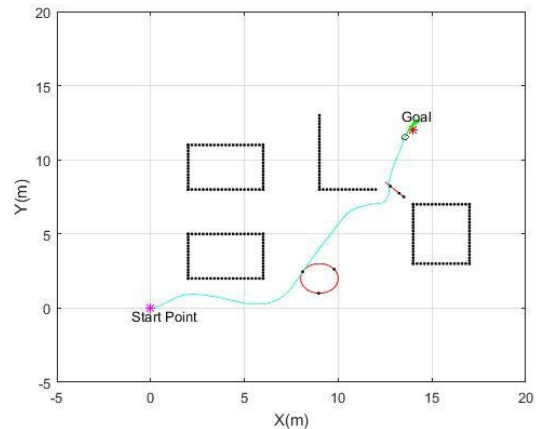
B. EXPERIMENTAL RESULTS

1) EXPERIMENTAL SETUP

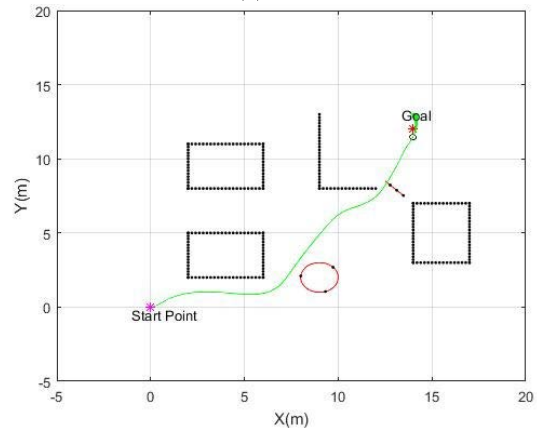
In this section, the proposed FIDWA controller is validated experimentally. The proposed approach performance is evaluated in several static and dynamic environments. In these experiments the Qbot open-architecture ground



(a) DWA



(b) FDWA



(c) FIDWA

FIGURE 10. Structured dynamic environment testing scenario.

robot from Quanser is used, this robot is built on a Kobuki differential mobile platform [27], [28], [29]. The robot is equipped with a Kinect sensor [28], [29] to detect the obstacles and is supported with real-time based controller under MATLAB/Simulink using QUARC software [29], [30]. The Kinect sensor utilizes the measured depth data with a range of 5m. Table 4 and figure 12, show the physical properties of the robot. The mobile robot uses odometric localization using its encoder sensors to localize itself within the map of the robot

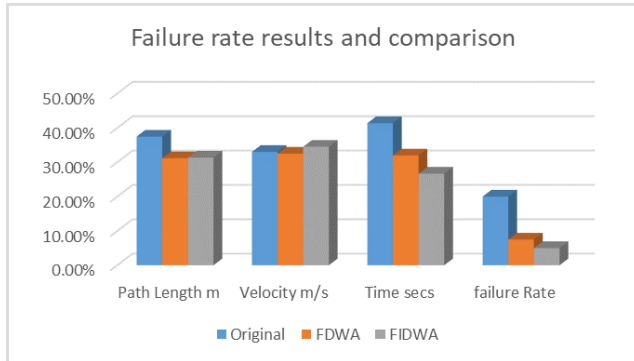


FIGURE 11. Algorithms failure rate results.

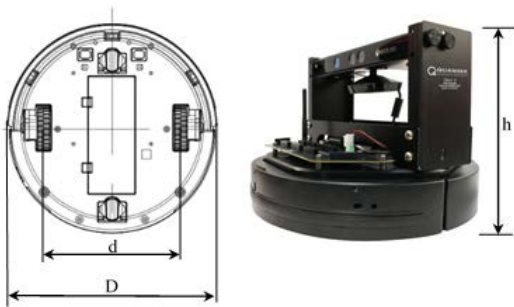


FIGURE 12. Quanser differential mobile robot [27].

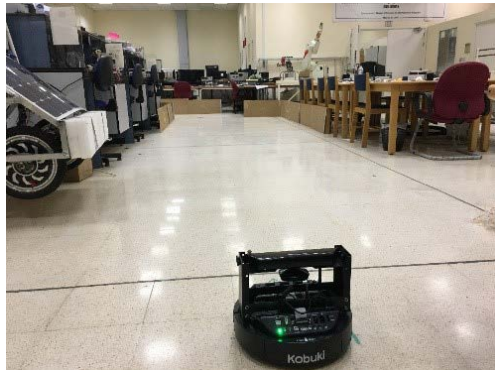


FIGURE 13. Experimental environment snapshot.

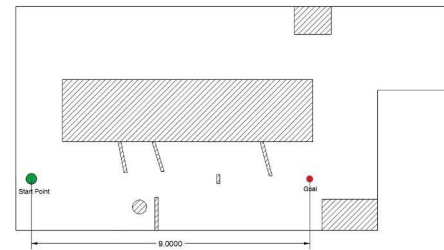
surrounding environment. All the results in this section are collected directly by the robot odometric localization module. Figure 13 shows a snapshot from the laboratory where all experiments were conducted.

2) EXPERIMENTAL VALIDATION IN STATIC ENVIRONMENT

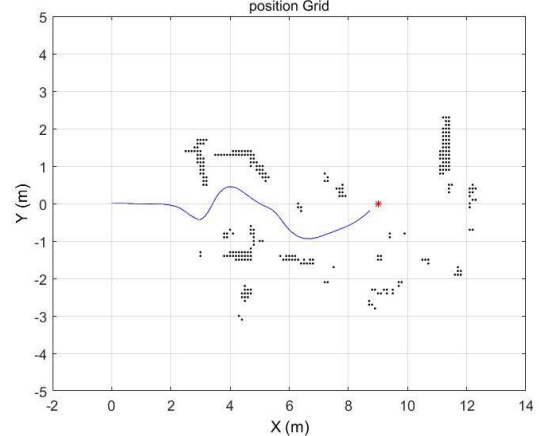
Firstly, the FIDWA was tested in an environment with multiple static obstacles as shown in figure 14a. In this experiment, the robot was designed to reach the goal point located in a straight line ahead of the robot while avoiding six static obstacles. The robot was able to reach the goal without hitting any of the static obstacles. Figure 14b shows the results collected from the robot and the map drawn by the Kinect sensor. Figure 15 shows the velocities of the robot during the static obstacles test, while figure 16 shows the change in the weight parameters during the same test.

TABLE 4. Quanser differential mobile robot specifications.

Symbol	Description	Value	Unit
D	Diameter	0.35	m
d	Distance between the wheels	0.235	m
h	Hight with Kinect mounted	0.27	m
$v_{max}$	Maximum speed	0.7	m/s
m	Mass	3.79	kg



(a) Test area



(b) Test result

FIGURE 14. Multiple static obstacles testing scenario.

3) EXPERIMENTAL VALIDATION IN DYNAMIC ENVIRONMENT

Next, the FIDWA was tested in an environment with one dynamic obstacle and a small corridor as shown in figure 17a. In this experiment, the environment included several static obstacle and one dynamic obstacle moving in a circular motion. The robot is supposed to avoid all static and dynamic obstacles and move around the corridor to reach the goal point. The robot was able to avoid the obstacle successfully and reach the goal location as shown in figure 17b. Figures 18 and 19 show the robot velocities and the change in the weight parameters appropriately during the test.

4) EXPERIMENTAL VALIDATION WITH SEVERAL WAY-POINTS ENVIRONMENT

Lastly, the algorithm was tested with several way-points as shown in figure 20. In this experiment, the robot had more



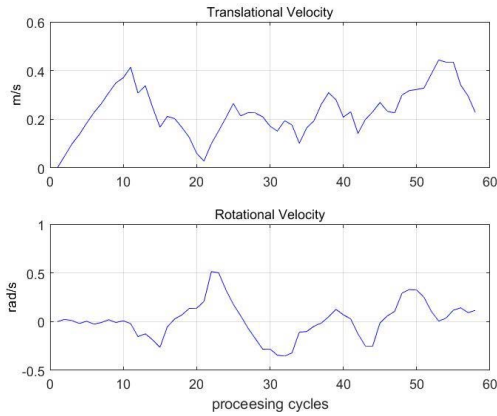


FIGURE 15. Static obstacles test velocities.

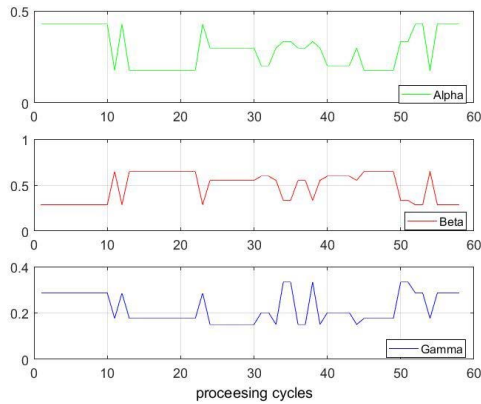


FIGURE 16. Static obstacles test weights.

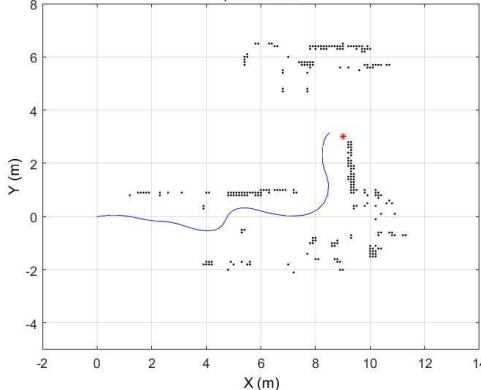
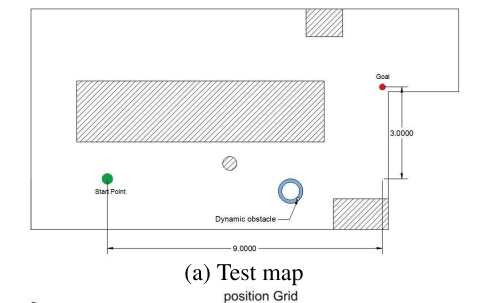


FIGURE 17. Dynamic obstacles testing scenario.

than three goals to reach. First goal (W1) is a head of the robot with dynamic and static obstacles. Second goal (W2)

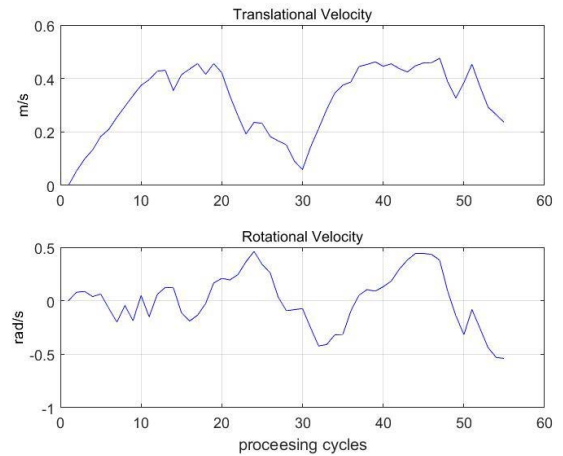


FIGURE 18. Dynamic obstacles test velocities.

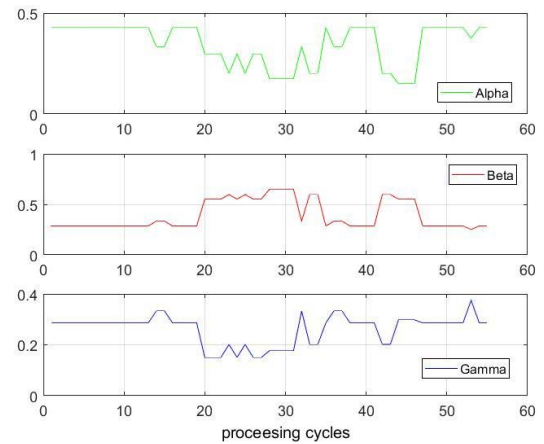


FIGURE 19. Dynamic obstacles test weights.

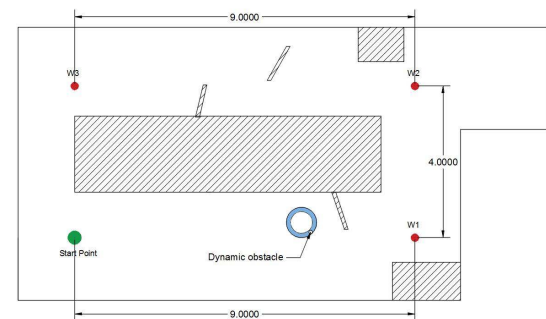


FIGURE 20. Way-points Test map testing scenario.

is on the side of the first goal where the robot is supposed to change its heading to reach the goal. Third goal (W3) is at the same distance of the starting point of the robot with some static obstacles on the way. The robot was able to reach each goal successfully while avoiding all the obstacles in its way as can be seen from figure 21. This test proves the possibility of using this controller with another global path planning algorithm to avoid obstacles between way-points calculated by the path planning algorithm. Figure 22 shows the velocities of the

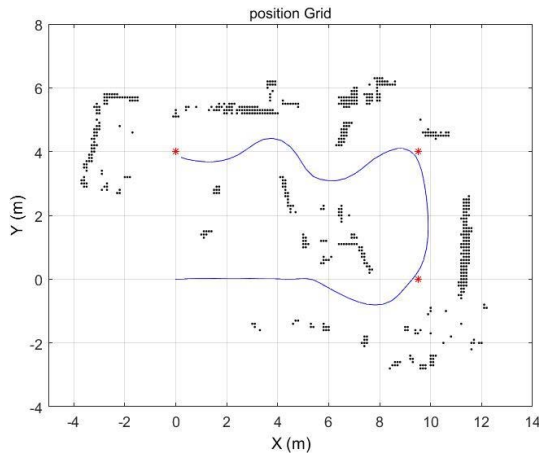


FIGURE 21. Way-points test map drawn by the robot.

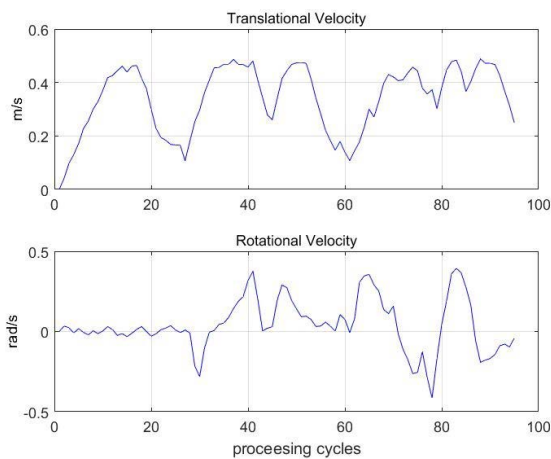


FIGURE 22. Way-points test robot velocities.

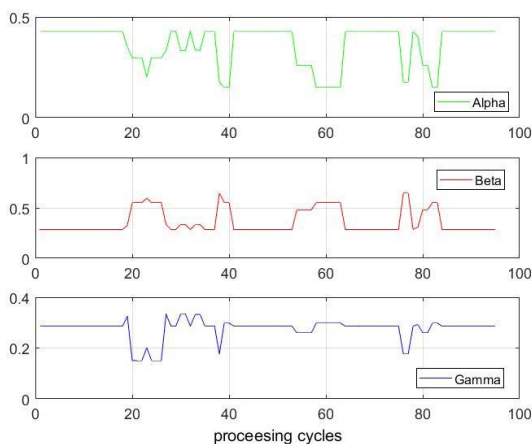


FIGURE 23. Way-points test weights.

robot during the way-points test, and figure 23 demonstrates the objective function weights changes within the same test.

## VI. CONCLUSION

In this paper, a new optimization technique for dynamic window approach is presented. The DWA is based on the

kinematic model of the robot. One of the main problems facing the dynamic window approach is the selection of the weights of its objective function. The proposed technique depends on the physical properties of the environment around the robot. The algorithm then uses a fuzzy logic controller to update the weight parameters of the objective function. The FLC inputs are the distances to the nearest obstacles in five areas in front of the robot, while its outputs are the weight parameters.

The proposed method is compared with the other methods the basic DWA and FDWA in both static and dynamic environments. The algorithms are compared using three different criteria: the path length, the average linear velocity, and the time required to reach the goal. Real experiments were conducted to validate the performance of the FIWDA.

The results show that the proposed technique can considerably improve the performance of the original DWA algorithm in terms of success rate to reach the goal point, the path length, the average velocity, and also the time required, with very minimal sacrifices to processing power required compared to the original DWA.

## ACKNOWLEDGMENT

This article represents the opinions of the author(s) and does not mean to represent the position or opinions of the American University of Sharjah.

## REFERENCES

- [1] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991, doi: [10.1109/70.88137](https://doi.org/10.1109/70.88137).
- [2] N. Y. Ko and R. G. Simmons, "The lane-curvature method for local obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Oct. 1998, pp. 1615–1621, doi: [10.1109/IROS.1998.724829](https://doi.org/10.1109/IROS.1998.724829).
- [3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: [10.1109/100.580977](https://doi.org/10.1109/100.580977).
- [4] M. Kulich, V. Kozák, and L. Přeučil, "Comparison of local planning algorithms for mobile robots," in *Modelling and Simulation for Autonomous Systems*, J. Hodický, Ed. Cham, Switzerland: Springer, 2015, pp. 196–208.
- [5] E. J. Molinos, Á. Llamazares, and M. Ocaña, "Dynamic window based approaches for avoiding obstacles in moving," *Robot. Auton. Syst.*, vol. 118, pp. 112–130, Aug. 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889018309746>
- [6] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field," *Soft Comput.*, vol. 16, no. 1, pp. 153–164, 2012, doi: [10.1007/s00500-011-0742-z](https://doi.org/10.1007/s00500-011-0742-z).
- [7] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Jan. 2011, doi: [10.1109/TIE.2011.2109332](https://doi.org/10.1109/TIE.2011.2109332).
- [8] M. J. Er and C. Deng, "Obstacle avoidance of a mobile robot using hybrid learning approach," *IEEE Trans. Ind. Electron.*, vol. 52, no. 3, pp. 898–905, Jun. 2005, doi: [10.1109/TIE.2005.847576](https://doi.org/10.1109/TIE.2005.847576).
- [9] P. Nattharith and M. S. Güzel, "Machine vision and fuzzy logic-based navigation control of a goal-oriented mobile robot," *Adapt. Behav.*, vol. 24, no. 3, pp. 168–180, Jun. 2016, doi: [10.1177/1059712316645845](https://doi.org/10.1177/1059712316645845).
- [10] P. Nattharith, "Fuzzy logic based control of mobile robot navigation: A case study on iRobot Roomba platform," *Sci. Res. Essays*, vol. 8, pp. 82–94, Jan. 2013.
- [11] M. A. K. Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robot. Comput. Integr. Manuf.*, vol. 27, no. 1, pp. 135–149, Feb. 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584510000700>

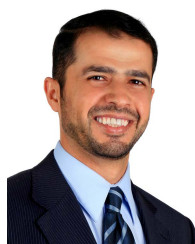
- [12] W. Zhu and M. Hayashibe, "A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation," *IEEE Trans. Ind. Electron.*, early access, Jul. 20, 2022, doi: [10.1109/TIE.2022.3190850](https://doi.org/10.1109/TIE.2022.3190850).
- [13] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 1999, pp. 341–346, doi: [10.1109/ROBOT.1999.770002](https://doi.org/10.1109/ROBOT.1999.770002).
- [14] H. Berti, A. Sappa, and O. Agamennoni, "Improved dynamic window approach by using Lyapunov stability criteria," *Latin Amer. Appl. Res.*, vol. 38, no. 1, pp. 289–298, 2008.
- [15] C.-C. Chou, F.-L. Lian, and C.-C. Wang, "Characterizing indoor environment for robot navigation using velocity space approach with region analysis and look-ahead verification," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 2, pp. 442–451, Feb. 2011, doi: [10.1109/TIM.2010.2058531](https://doi.org/10.1109/TIM.2010.2058531).
- [16] X. Li, F. Liu, J. Liu, and S. Liang, "Obstacle avoidance for mobile robot based on improved dynamic window approach," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 25, pp. 666–676, Jan. 2017, doi: [10.3906/elk-1504-194](https://doi.org/10.3906/elk-1504-194).
- [17] Z. Hong, S. Chun-Long, Z. Zi-Jun, A. Wei, Z. De-Qiang, and W. Jing-Jing, "A modified dynamic window approach to obstacle avoidance combined with fuzzy logic," in *Proc. 14th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Aug. 2015, pp. 523–526, doi: [10.1109/DCABES.2015.136](https://doi.org/10.1109/DCABES.2015.136).
- [18] O. A. Abubakr, M. A. K. Jaradat, and M. A. Hafez, "A reduced cascaded fuzzy logic controller for dynamic window weights optimization," in *Proc. IEEE 11th Int. Symp. Mechatronics Appl. (ISMA)*, Mar. 2018, pp. 1–4, doi: [10.1109/ISMA.2018.8330141](https://doi.org/10.1109/ISMA.2018.8330141).
- [19] P. Saranrittichai, N. Niparnan, and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on dynamic window approach," in *Proc. 10th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2013, pp. 1–4.
- [20] D. Lima and A. Victorino, "An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments," in *Proc. IEEE ICRA Workshop Modelling, Estimation, Perception Control Terrain Mobile Robots*, May 2014, pp. 1–7.
- [21] B. Choi, B. Kim, E. Kim, and K. Yang, "A modified dynamic window approach in crowded indoor environment for intelligent transport robot," in *Proc. 12th Int. Conf. Control, Automat. Syst.*, Oct. 2012, pp. 1007–1009.
- [22] D. A. de Lima and G. A. S. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *J. Control, Autom. Electr. Syst.*, vol. 24, nos. 1–2, pp. 106–116, Apr. 2013, doi: [10.1007/s40313-013-9006-5](https://doi.org/10.1007/s40313-013-9006-5).
- [23] Y. Kang, D. A. de Lima, and A. C. Victorino, "An approach of human driving behavior correction based on dynamic window approach," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 304–309.
- [24] Y. Kang and A. C. Victorino, "Human-vehicle cooperative driving using image-based dynamic window approach: System design and simulation," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2487–2492.
- [25] Y. Wang, W. She, M. Fu, F. Ding, and S. Dai, "Real-time obstacle avoidance of hovercraft based on follow the gap with dynamic window approach," in *Proc. OCEANS MTS/IEEE Charleston*, Oct. 2018, pp. 1–8, doi: [10.1109/OCEANS.2018.8604720](https://doi.org/10.1109/OCEANS.2018.8604720).
- [26] Z. Chen, Y. Zhang, Y. Zhang, Y. Nie, J. Tang, and S. Zhu, "A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles," *IEEE Access*, vol. 7, pp. 126439–126449, 2019, doi: [10.1109/ACCESS.2019.2936689](https://doi.org/10.1109/ACCESS.2019.2936689).
- [27] Y. Robot. (2015). *Kobuki Mobile Robot*. Accessed: Mar. 15, 2018. [Online]. Available: <http://kobuki.yujinrobot.com/about2/>
- [28] Microsoft. (2012). *Kinect Sensor Manual*. Accessed: Apr. 13, 2018. [Online]. Available: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>
- [29] Microsoft. (2015). *Qbot 2 for Quarc*. Accessed: Apr. 13, 2018. [Online]. Available: <http://kobuki.yujinrobot.com/qbot-2-for-quarc/>
- [30] H. Ghaderi, M. Yadegar, N. Meskin, and M. Noorizadeh, "A novel proportional navigation based method for robotic interception planning with final velocity control," *IEEE Access*, vol. 9, pp. 106428–106440, 2021, doi: [10.1109/ACCESS.2021.3091152](https://doi.org/10.1109/ACCESS.2021.3091152).
- [31] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Studies*, vol. 7, pp. 1–13, Jan. 1975. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020737375800022>
- [32] J. F. Baldwin and B. W. Pilsworth, "Axiomatic approach to implication for approximate reasoning with fuzzy logic," *Fuzzy Sets Syst.*, vol. 3, no. 2, pp. 193–219, Mar. 1980. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0165011480900548>
- [33] W. Van Leekwijck and E. E. Kerre, "Defuzzification: Criteria and classification," *Fuzzy Sets Syst.*, vol. 108, no. 2, pp. 159–178, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165011497003370>



**OMER ALI ABUBAKR** received the B.S. degree in electrical and electronics engineering with a minor in electronics systems software engineering from the University of Khartoum, Khartoum, Sudan, in 2014, and the M.S. degree in mechatronics engineering from the American University of Sharjah, Sharjah, UAE, in 2018. His current research interests include mechatronics, robotics, computer vision, and machine learning.



**MOHAMMAD A. JARADAT** received the master's and Ph.D. degrees in mechanical engineering from Texas A&M University, College Station, TX, USA. He is currently with the American University of Sharjah (AUS) and the Jordan University of Science and Technology (JUST) in the Mechanical Engineering Department. His scientific experience has resulted in many publications, patents, projects, and awards specialized in the following research areas: robotics, artificial intelligent systems, sensor fusion, fault diagnostics, intelligent NANO-systems, intelligent control, mechatronics system design, and control systems.



**MAMOUN F. ABDEL-HAFEZ** (Senior Member, IEEE) received the B.S. degree from the Jordan University of Science and Technology, in 1997, the M.S. degree from the University of Wisconsin, Milwaukee, in 1999, and the Ph.D. degree from the University of California at Los Angeles (UCLA), in 2003, all in mechanical engineering. He was a Postdoctoral Research Associate with the Department of Mechanical and Aerospace Engineering, UCLA, in 2003, where he was involved in a research project on fault-tolerant autonomous multiple aircraft landing. He is currently with the Department of Mechanical Engineering, American University of Sharjah. His research interests are in stochastic estimation, control systems, sensor fusion, and fault detection.

...