# A Reduced Cascaded Fuzzy Logic Controller for Dynamic Window Weights Optimization

**Omer Ali Abubakr , Mohammed Abdel Kareem Jaradat * and Mamoun Adel Hafez ***

Mechatronics Engineering

* Department of Mechanical Engineering

American University of Sharjah, Sharjah, UAE

***Abstract*— Mobile robots in indoor environments require path planning techniques with very fast processing cycles and ability to work in dynamic environments. The dynamic window approach is one of the most sophisticated approaches in obstacle avoidance. It uses objective function to choose the optimal velocity commands. The objective function consist of three components. Changing the weights of these component will change the behavior of the algorithm. In this paper, a fuzzy logic controller that depends only on the obstacles distribution around the robot and uses less number of rules in the fuzzy logic is proposed. When compared with the other fuzzy logic controller this approach is much faster, reducing the processing cycle time for the overall approach and also the path obtained is smoother and with higher velocities in areas crowded with obstacles.**

***Keywords— Path Planning, Obstacle avoidance, DWA, Fuzzy Logic***

## I. INTRODUCTION

The main objective of any autonomous mobile robot is to reach its destination point in indoor or outdoor environments using the shortest possible path and without colliding with any obstacles that might exist in its environment. Several path planning techniques were developed that control the motion of the robot from its starting point to its destination. One example of path planning techniques is the 'Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field' proposed by Jaradat et el. In 2012[1]. These methods rely on static model or a map of the environment and hence they seize to function in unknown environments or where the obstacles are moving within the map. Obstacle avoidance techniques have been developed to enable robots to reach their goals in unknown and dynamic environments such as in [2]. One of the most famous obstacle avoidance techniques is the dynamic window approach 'DWA'[3].

One of the advantages of the dynamic window approach is the consideration of kinematic constraints of the robot which enables the robot to avoid collision with dynamic as well as static obstacles that intersects with the robot path towards the goal.

978-1-5386-1078-7/18/$31.00 ©2018 IEEE

DWA was first proposed by D. Fox in 1997. It takes into account the kinematic and dynamic constraints of a mobile robot.

The 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, ω) of the translational velocity v and rotational velocities ω that are achievable by the robot [4]. The algorithm uses the velocity space to search for the velocity command (v, ω) that maximizes the objective function of the algorithm.

Several extensions were proposed to the original DWA that should enhance the performance of the algorithm. In 1999, Brock and Khatib presented the Global Dynamic window approach[4], its objective was to ensure a path towards the goal avoiding local minima that faced the original DWA. The algorithm was combined with local minima free navigation function NF1 to ensure that the robot does not stuck in a local minima when selecting the next velocity command. Arras et al.[5] presented the reduced-DWA to speed up the translational velocity selection by quantizing the shift pointer of the DWA circuit. The reduced-DWA requires much less processing power but it is not appropriate when the robot is not oriented towards the goal and the goal heading is more the 90 degrees. In 2008, H. BERTI et al. presented an improved dynamic window approach using Lyapunov stability criteria [6]. They proposed a new objective function that includes Lyapunov stability criteria. It guarantees a global and asymptotic convergence to the goal avoiding collisions and resulting in a more simple and self-contained approach. Chih-Chung Chou et al introduced a new enhancement on the dynamic window approach called DWA* in 2011[7]. In this method, a candidate velocity is derived by analyzing the intervals to find navigable areas in the environment. The method uses search trees to evaluate the intervals and decide the next motion command. In 2015, . Zhang Hong et el.[8], suggested using fuzzy logic to update the weights of the objective function of the DWA. Their objective is to update these weights using information from the robot environment that enhance the performance of the original DWA and its extensions.

In this paper, the fuzzy logic controller for the weight parameters change is updated to enhance the performance of the controller and reduce its complexity. The proposed method uses information about the obstacles only to update the objective function weights.

This paper is organized as follows. Section II gives a brief introduction to the original DWA algorithm and the previous approach of the fuzzy logic design. Section III, gives the details of the proposed modifications on the fuzzy logic design. Simulation results and comparison are shown in section IV. Section V concludes the proposed method.

## II. DWA AND FUZZY OPTIMIZATION

### *A. The dynamic window approach*

As mentioned above the dynamic window approach is an obstacle avoidance technique that is proposed by D. Fox [1]. Its main advantage is that it considers the kinematic constraints of the robot. This is done by searching a well-chosen velocity space. As explained before, the velocity space is all possible sets of tuples (v, ω) where v is the velocity and ω is the angular velocity. The approach assumes that robots move only on circular arcs representing each such tuple, at least during one timestamp. This space is bounded by constraints affecting the robot's behavior; some of those constraints are imposed by the obstacles in the environment, while others are from the kinematic limitations of the robot (maximum velocity and acceleration). The main concept of the algorithm is to get an optimal velocity command from the velocity space by maximizing the objective function. The velocity command is gained from the configuration space of the robot. This space includes three kinds of velocities spaces. The first velocity space contains all the possible velocities that the robot can generally achieve and is called $V_s$. In the second velocity space each velocity ensures that the robot stops before colliding with any obstacle if the corresponding velocity is chosen as the velocity command $V_a$. The third velocity space contains only those velocities that can be reached within the next processing cycle due to the acceleration limitations and is called the dynamic window $V_d$. The resultant velocity search space is given by $V_r$ which can be seen in figure-1 as the white area inside $V_d$ and is given by the equation:

$$V_r = V_s \cap V_a \cap V_d \quad (1)$$

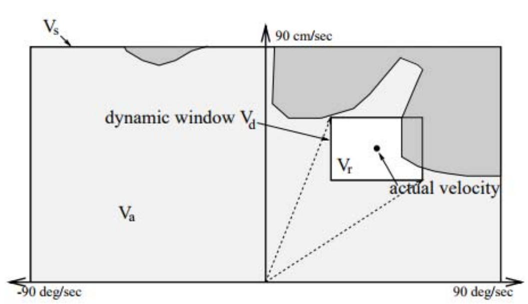



Figure 1 - The dynamic window [3]

Once the resultant velocity space is obtained the most appropriate velocity command is picked. Each velocity tuple (v, ω) is evaluated with the objective function. In the original DWA the objective function is defined as follows:

$$G(v,\omega) = \alpha.\ \mathrm{Heading}(v,\omega) + \beta.\ \mathrm{dist}(v,\omega) + \gamma\ .\ \mathrm{velocity}(v,\omega) \quad (2)$$

Where α, β and γ are constants and their summation is equal to 1. In the objective function, the heading is the angle between the robot heading angle and the goal position, while the dist. component measures the closeness of the robot with the closest obstacle for the sampled velocity. The velocity is used to force the robot to move towards the goal as fast as possible. Finally, the velocity command with the highest value of the objective function is selected as the next motion command.

### *B. Fuzzy Logic weight optimization*

In the DWA, using constant weights for the objective function might not be practical for all the cases. Especially if the obstacles are dynamic or the exist very near to goal position. Slight changes on the weights might improve the approach. Zhang Hong et el.[7] suggested using fuzzy logic to change the parameters weight. The basic idea is to update the weights based on the distribution of obstacles around the robot, the target position, and the target orientation (θ). The algorithm uses a fuzzy logic controller to find the obstacle distribution type based on the existence of the obstacle in three areas around the robot: left, front, and right. Figure-2 shows all possible combinations of the distribution types. The output of this controller is one of the eight types shown in the figure.

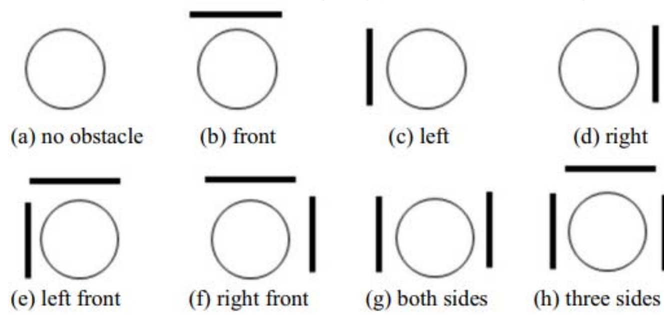



Figure 2 - Obstacle distribution type [8]

The algorithm then calculates the distant to the target and the target heading. The three parameters are then used in a second controller to update the weight parameters α, β and γ of the objective function.

## III. THE PROPOSED METHOD

The fuzzy logic controller is used to update the weights of the three components of the objective function in real time. This leads to an optimized DWA that is able to change its priorities according to its environment. In this paper, the controller is updated to use only the obstacle distribution type in three areas around the robot 'left, front and right' and neglecting the target position and orientation when compared to [8]. The result is a reduced fuzzy logic controller with less number of inputs and hence less number of rules. The processing cycle of the overall system is greatly reduced due to these updates. The procedure of the proposed method is shown in figure-3.

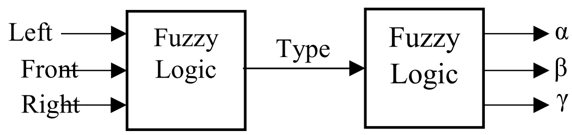



Figure 3 - The proposed Method

As explained before, the output of the first fuzzy controller is one of the eight distribution types as shown in figure-2. The second fuzzy controller uses this type to update the weight parameters based on a set of predefined rules. In this case, the number of rules is equal to the number of distribution types. The fuzzy rules can be set as follows:

If type is a then α is VH and β is VS and γ is S.
If type is b then α is M and β is H and γ is S.
If type is c then α is VH and β is S and γ is M.
If type is d then α is H and β is M and γ is S.
If type is e then α is H and β is S and γ is S.
If type is f then α is H and β is H and γ is S.
If type is g then α is H and β is M and γ is S.
If type is h then α is M and β is H and γ is VS.

Figure 4 - Fuzzy rules of the proposed method

Mamdani inference engine is used to do the fuzzy reasoning with triangular membership functions for the inputs and gaussian bell membership functions for the outputs for both controllers as shown in figure-5 and figure-6. The output parameters of the fuzzy logic is then normalized to ensure that $\alpha + \beta + \gamma = 1$.

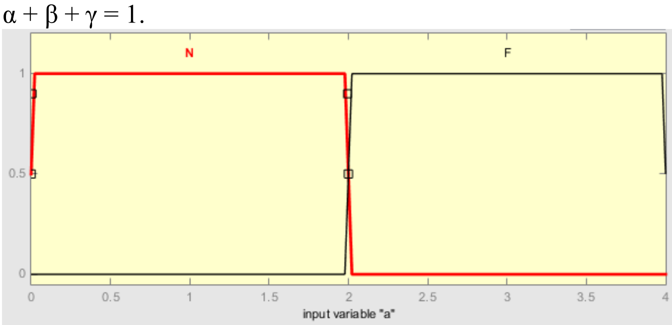



Figure 5- Membership functions of the fuzzy controller inputs

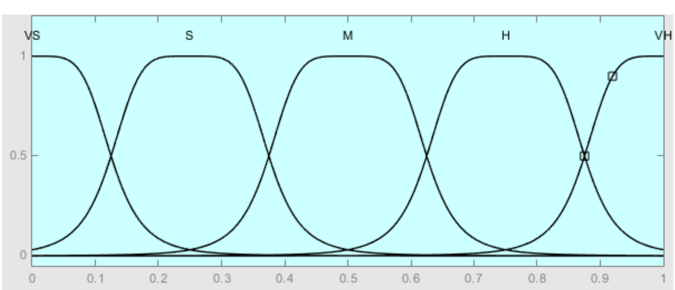
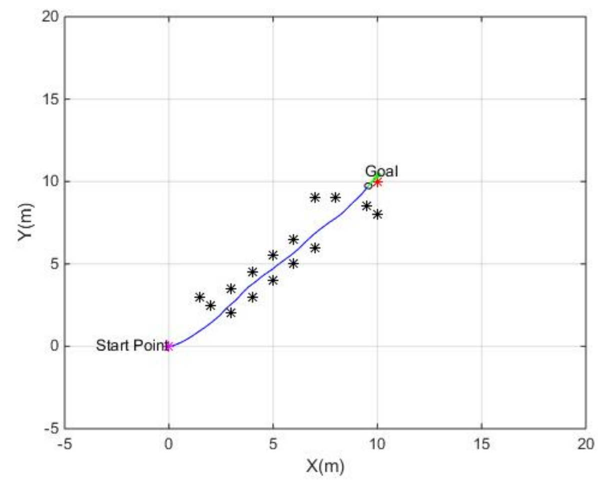



Figure 6- Membership functions of the fuzzy controller outputs

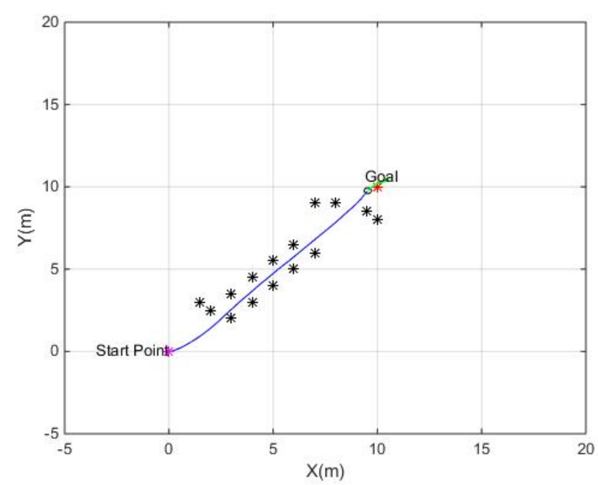
## IV. Simulation Results

In this section, a comparison between the previously proposed fuzzy logic controller for weight optimization and the reduced fuzzy logic controller is presented. The original DWA was simulated in Matlab. The maximum translational velocity is 1 m/s and the translational acceleration is 0.2 m/s$^2$. The maximum rotational velocity and acceleration are 20°/s and 50°/s$^2$ respectively. The robot has a circular shape with a radius of 0.35 m.

The two approaches were tested in different scenarios and the results are compared in terms of the path length, the average velocity, and the time it takes to reach the goal. Figure-7 shows a sample test case were the two approaches are compared in an environment with point obstacles that simulates a corridor. Figure-8 compares the two approaches in an environment with both static and dynamic obstacles. The dynamic obstacles used are points moving in both circular and linear paths as shown in the diagram with circular and linear paths shown in red.

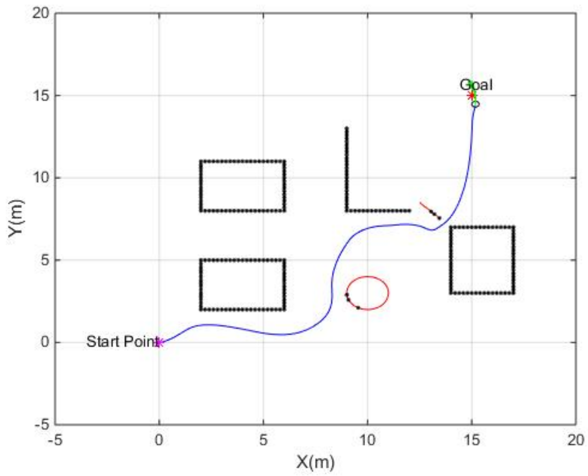



(a) DWA with the original fuzzy controller

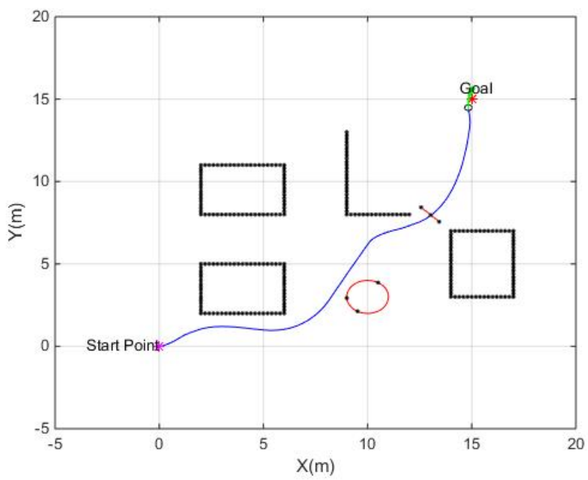



(b) DWA with the reduced fuzzy controller

Figure 7 - Simulation result with point obstacles




(a) DWA with the original fuzzy controller




(b) DWA with the reduced fuzzy controller

Figure 8- Simulation result with dynamic obstacles

Table-1 shows the results of both tests. The two algorithms are compared and the original controller is shown in column 'O' while the proposed controller is shown in column 'P'.

Table 1- Test results comparison

| | Path Length (m) | | Average Velocity (m/s) | | Time (seconds) | |
|---|---|---|---|---|---|---|
| | O | P | O | P | O | P |
| Test 1 | 13.8 | 13.8 | 0.41 | 0.5977 | 46.5 | 31.75 |
| Test 2 | 26.1 | 23.6 | 0.65 | 0.73 | 85.23 | 66.46 |

From the table, the time required to reach the goal was reduced when using the proposed controller when compared to the original controller, also the velocities were higher and the path in some cases was shorter.

## V. Conclusion

In this paper, a reduced fuzzy logic controller is proposed to do the weight optimization on the dynamic window approach for obstacle avoidance in mobile robots. The weights of the objective function are modified in real time depending only on the distribution of obstacles around the robot. The simulation results show the improvement in the performance when compared with the original approach.

## References

[1] Jaradat, Mohammad Abdel Kareem, Mohammad H. Garibeh, and Eyad A. Feilat. "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field." Soft Computing 16, no. 1 (2012): 153-164.

[2] Jaradat, Mohammad Abdel Kareem, Mohammad Al-Rousan, and Lara Quadan. "Reinforcement based mobile robot navigation in dynamic environment." Robotics and Computer-Integrated Manufacturing 27, no. 1 (2011): 135-149.

[3] D. Fox, W. Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, 1997.

[4] O. K. Oliver Brock, "High Speed Navigation Using the Global Dynamic Window Approach," presented at the International Conference on Robotics & Automation, Detroit. Michigan, 1999.

[5] Arras, K.O., J. Persson, N. Tomatis and R. Siegwart, "Real-Time Obstacle Avoidance For Polygonal Robots With A Reduced Dynamic Window," IEEE International Conference on Robotics and Automation, 3050- 3055 (2002).

[6] H. Berti, A.D. Sappa and O.E. Agamennoni "MPROVED DYNAMIC WINDOW APPROACH BY USING LYAPUNOV STABILITY CRITERIA,"

[7] 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 TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, vol. 60, 2011.

[8] H. Zhang, C.-l. Sun, Z.-j. Zheng, W. An, D.-q. Zhou, and J.-j. Wu, "A modified Dynamic Window Approach to Obstacle Avoidance combined with Fuzzy Logic," presented at the Distributed Computing and Applications for Business Engineering and Science (DCABES),, Guiyang, China, 2015.

.