



Article

# Predictive Dynamic Window Approach Development with Artificial Neural Fuzzy Inference Improvement

Daniel Teso-Fz-Betoño <sup>1,\*</sup> , Ekaitz Zulueta <sup>1</sup>, Unai Fernandez-Gamiz <sup>2</sup> , Aitor Saenz-Aguirre <sup>1</sup> and Raquel Martinez <sup>1</sup>

<sup>1</sup> System Engineering and Automation Control Department, University of the Basque Country (UPV/EHU), Nieves Cano, 12, 01006 Vitoria-Gasteiz, Spain

<sup>2</sup> Department of Nuclear and Fluid Mechanics, University of the Basque Country (UPV/EHU), Nieves Cano, 12, 01006 Vitoria-Gasteiz, Spain

\* Correspondence: daniel.teso@ehu.eus

Received: 27 June 2019; Accepted: 23 August 2019; Published: 26 August 2019



**Abstract:** The aim of this paper is to improve the dynamic window approach algorithm for mobile robots by implementing a prediction window with a fuzzy inference system to adapt to fixed parameters, depending on the surrounding conditions. The first implementation shows the advantage of the prediction step in terms of optimizing the path selection. The second improvement uses fuzzy inference to optimize each of the fixed parameters' values to increase the algorithm performance. Nevertheless, a simple fuzzy inference system (FIS) was not used for this particular study; instead, an artificial neuro-fuzzy inference system (ANFIS) was used, thus making it possible to develop a FIS system with a back-propagation technique. Each parameter would have a particular ANFIS, in order to modify the  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  parameters individually. At the end of the article, different scenarios are analyzed to determine whether the developments in this article have improved the DWA behavior. The results show that the prediction step and ANFIS adapt DWA performance by optimizing the path resolution.

**Keywords:** DWA; ANFIS; motion planning; mobile robots; obstacle avoidance; fuzzy logic; MPC

## 1. Introduction

Presently, mobile robots are becoming more intelligent and can move autonomously in unknown environments. Mapping, localization, path-planning, and path-following aspects have to be implemented in order to solve this inconvenience. Jin et al. [1] remarked that the smart movement of the robot is associated with the intelligence that has been implemented. Hence, these intelligences are improved by using a combination of sensors and information processing algorithms, as Ortigoza et al. [2] explained.

Local path-planning or path-following algorithms autonomously move automated guided vehicles (AGVs) to their goal position. Thus, this problem can be described as the process of determining a suitable and safe path, avoiding collisions with the obstacles [3]. There are many well-known methods to implement this task, such as the dynamic window approach (DWA), which calculates a potential path considering speed restrictions, as Fox et al. [4] explained. A vector field histogram was developed with the aim of being computationally efficient and robust. In practice, the algorithm became fast and reliable in populated obstacle scenarios, as Borenstein et al. [5] related. The TangentBug and PointsBug algorithms are really popular, and were compared by Buniyamin et al. [6]. The TangentBug algorithm depends on the range of the sensor, which is mounted on the robot. Hence, the action plan to reach to the goal uses the sensor information and the knowledge of the robot current position and goal position.

However, PointsBug depends on the sensor range, the sudden point, and the trigonometric function. Thus, the algorithm calculates where the next point is to move from the starting point to the goal.

Apart from these classic functions for motion planning, there are some more modern algorithms, for example, swarm optimization (SO), as Ever [7] has shown. This method is based on a population of candidate solutions that are moving around the search space. The movement of these particles is influenced by the best local particle position. These aspects move the swarm toward the best solutions, which is the goal position. Dynamic movement primitives (DMP), as Mei et al. [8] analyzed, are another solution, which were motivated by the desire to find a way to represent complex motor actions that can be flexibly adjusted without manual parameter tuning or having to worry about instability. On the end, this intelligence tries to simplify complex movements on a set of primitive actions. Apart from DMP, there is also moving to a point with obstacle avoidance, which uses simple mathematical equations to avoid obstacles, and is proposed by Teso-Fz-Betoño et al. [9]. Furthermore, it is important to consider Garcia et al. [10], who proposed a method for industrial mobile robots, which it improve the trajectory in dynamic environments, applying an elastic-band-based method in combination with a model predictive trajectory planner. Apart from this, Pfeiffer et al. [11] described another investigation, which uses a supervised model to learn complex mapping, and processes the steering commands to arrive at the goal position. Moreover, a model predictive control (MPC), as Yu et al. [12] proposed, is another intelligence for the vehicles, which provides vehicle control to track the path. With the same technique, Qingjia et al. [13] improved the results by adopting multi-constraints model predictive control (MMPC). Genetic algorithms are also implemented in mobile robots. Sathiya et al. [14] explained a novel concept by adopting two techniques, namely multi-objective differential evolution and non-dominated sorting genetic algorithm (NSGA-II), in order to obtain a safer path by optimizing travel time and actuators effort. The ant algorithm is implemented in navigation problematics, such as Xiaoping et al. [15] explained. However, the recent investigation has been improved by adopting different techniques. For example, Dai et al. [16] implemented the A\* characteristics into an ant algorithm and Yang et al. [17] presented an efficient double layer ant colony, which is called DL-ACO. This novel concept uses two ant colony running independently and successively. First one controls a collision free path and then the second one optimizes the results of the first ant colony trajectory.

As Bellotto et al. [18] and Truong et al. [19] analyzed, there are some situations where the AGV has to interact with humans. This is the reason algorithms must change their behavior, performing harmless maneuvering depending on the scenario conditions.

In the end, the motion planning algorithms have to adapt their behavior with dynamic surroundings. Thus, it is essential to implement an intelligence that changes these parameters from motion planning algorithms. Some investigations apply new solutions to modify DWA constants, depending on the area conditions. In the past, Hong et al. [20] adapted DWA using a fuzzy logic controller, achieving new suitable weights. These weights are modified considering the goal position and obstacles position information. Abubakr et al. [21] adopted the same controller; however, the weights were adjusted with obstacle distribution information.

Nevertheless, another alternative for the dynamic window approach is reduced-DWA, which requires much less processing power, as proposed by Arras et al. [22]. Abubakr et al. [21] analyzed a novel concept of reduce-DWA, applying a reduced cascaded fuzzy logic controller. The cascade uses two reduced fuzzy logic controllers. Hence, both controllers update three weights of DWA according to the environmental situation. The first controller inspects the surrounds of the robot and identifies the position of the obstacle for determining the obstacle distribution type. The second controller adopts the obstacle distribution type and assigns the best constant for a particular period time.

Overall, the DWA behavior can be adapted to the scenario conditions with fuzzy logic rules. In recent decades, however, fuzzy logic has been improved in learning capabilities, and is known as neuro-fuzzy, as Salleh [23] mentioned. In 1993, an adaptive neuro-fuzzy inference system (ANFIS) was introduced by Jang [24]. ANFIS has some limitations, such as the problem of dimensionality and training complexity, which restricts some applications with large datasets. Apart from that,

Zamani et al. [25] explained that ANFIS has a huge generalization capability compared with other neural networks or machine learning techniques. Furthermore, Badde et al. [26] compared the difference between fuzzy logic and ANFIS, concluding that ANFIS has better performance solving complex problems than the fuzzy controller.

Today, in mobile robots, ANFIS calculates the difference of angular velocity, adopting the smallest distance to the obstacles and the target angle, as Lu et al. [27] has shown. Singh et al. [28] investigated a solution for real-time navigation, in which ANFIS has to decide the optimal time for a collision free path in an unknown and dynamic environment. Imen et al. [29] presented a cascade controller. The first controller uses fuzzy logic, which identifies the curvature of the path and generates the input for the second controller, which is ANFIS. The second controller determines the path for the mobile robot. Pandey et al. [30] analyzed how ANFIS works as a controller for navigation and obstacle avoidance in unknown static environments.

There are different alternatives for path planning technique. The recent investigations try to develop classic algorithms to improve their performance. DWA algorithm that is a classic function, could be adopted another technique to improve the optimization of the path selection. DWA depends on the AGV model to solve a safe path, such as MPC. It seems logical to modify the optimization function, in order to analyze the future by adopting a prediction window like MPC does. Furthermore, this classic algorithm adopts different weights depending on the human criterion, hence, ANFIS, neural networks or fuzzy logic techniques can be adopted to optimize the fixed parameters. In Table 1 an overview of presented algorithms is made.

**Table 1.** An overview of presented algorithms.

Algorithm	Reference	Advantages	Weakness
Dynamic window approach	[4]	Consider robot constraints to calculate an optimum path.	Use fixed weights and select one path per sample time.
A vector field histogram	[5]	Places great emphasis on dealing with uncertainty from sensor and modeling errors.	Requires the kinematics and shape of the robot.
TangentBug	[6]	Produces very short paths given purely local information.	Need ideal localization Heuristic function may cause navigation to become slower.
PointsBug	[6]	Has better performance than TangentBug.	Require high accuracy localization based on output of trigonometric function.
Swarm optimization	[7]	Allows searching for an optimum trajectory by using particles in search space.	Requires a constant computing to optimize a solution.
Dynamic movements primitives	[8]	The complex movements are simplified by creating a set of primitive actions.	Requires updating all kernel weights and the quality of the path is compromised.
Moving to a point with obstacle avoidance	[9]	Simple equations are used to avoid obstacles and it does not need the kinematics equations for motion calculation.	Use fixed weights in the main function.
Model-based algorithms	[10–13]	The prediction step gives the chance to analyze the future and adjust the trajectory.	Require a perfect model of the plan to obtain a good prediction.
Genetic Algorithms	[14]	Normally uses the robot kinematic and dynamic parameters to optimize the path.	Uses a high computational capacity to optimize the function.
Ant colony optimization	[15]	Adopts a multiple agent to optimize the path resolution.	The probability distribution can change for each iteration and it is not optimized to real time, because of it requires uncertain time to converge.
Ant algorithm with A* characteristic	[16]	A* accelerates the ant colony conversion and increases the smoothness of global path.	A premature convergence could happen, because of bad adjustment of the limits.
Double Layer ant colony	[17]	Thanks to both executions the precision is increased.	The algorithms require a trajectory optimization to improve the path resolution.
Predictive DWA with ANFIS (Developed in this article)	-	Improves Fuzzy logic controller by adopting ANFIS technique to adjust DWA weights and foresee the future implementing a prediction.	Require a perfect model of the plan to obtain a good prediction and adjustment of the prediction step.

The main goal of the present study is to improve the DWA algorithm, not only by adapting each weight depending on the scenario circumstance, but also by implementing a prediction step to improve the optimization of the path selection. For the  $\alpha$ ,  $\beta$  and  $\gamma$  parameter calculations, despite using a classical fuzzy logic controller, ANFIS will be implemented. To make that possible, data collection will be made executing classical DWA with different weight configurations. The AGV speed, heading to the target and the distance to the obstacle will be evaluated in the data collection, so as to change the behavior of ANFIS to DWA by modifying these controller parameters to the scenario conditions.

## 2. Materials and Methods

DWA is a classical motion planning solution, as Fox et al. [4] analyzed. This algorithm is divided into two sections in order to predict the best trajectory. The first one is the search space, where different possible  $V_{agv}$  and  $\dot{\gamma}$  are considered. Moreover, this first section is divided into three subsections so as to filter these admissible variables, depending on the actual state. The first subsection is known as the circular trajectories, where the two-dimensional velocity  $(V_{agv}, \dot{\gamma})$  is calculated based on space. The second one takes all of the calculated velocities and selects only those admissible velocities that have a safe trajectory. The last subsection only contemplates these velocities that can be reached in a short time interval, considering the robot acceleration limitations.

In the second section, however, optimization is made by applying the target heading, distance to the obstacles, and actual velocity. This optimization just selects the best trajectory, rejecting other trajectories. This optimization, however, can be improved by adopting more than one path, as Kim et al. presented, such as in a confidence random tree-based algorithm [31]. In this article, this problem will be solved, applying a prediction step to the DWA algorithm, known as a predictive dynamic window approach (P-DWA).

Hence, to develop the P-DWA, it is important to place the basic aspect of the DWA optimization function, as Equation (1) represents and in Table 2 the DWA parameter and variable definition is represented.

$$G(V_{agv}, \dot{\gamma}) = \sigma(\alpha_D \cdot \text{heading}(V_{agv}, \dot{\gamma}) + \beta_D \cdot \text{dist}(V_{agv}, \dot{\gamma}) + \gamma_D \cdot \text{vel}(V_{agv}, \dot{\gamma})) \tag{1}$$

**Table 2.** Dynamic window approach (DWA) parameter and variable definition, as defined in the literature [4]. AGV—automated guided vehicle.

Description	Name	Units
Linear speed of vehicle	$V_{agv}$	m/s
Angular speed of vehicle	$\dot{\gamma}$	rad/s
Alignment of the robot with the target direction	$\text{heading}(V_{agv}, \dot{\gamma})$	-
Distance to obstacle	$\text{dist}(V_{agv}, \dot{\gamma})$	m
Displacement speed	$\text{vel}(V_{agv}, \dot{\gamma})$	m/s
Heading coefficient	$\alpha_D$	rad <sup>-1</sup>
Obstacle distance coefficient	$\beta_D$	m <sup>-1</sup>
Speed coefficient	$\gamma_D$	s/m
Fitness function	$G(V_{agv}, \dot{\gamma})$	-
Normalization function	$\sigma$	-

The heading is given by  $180^\circ - \theta$ , where  $\theta$  is the angle of the target point relative to the AGV heading direction as Figure 1 shown.

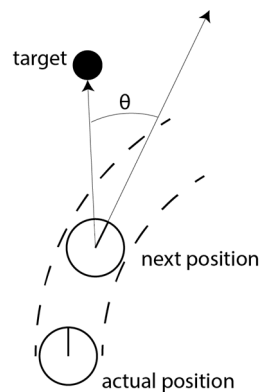


Figure 1. Angle  $\theta$  to the target.

Moreover,  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  are the constants that will be adapted using ANFIS. DWA does not always have the capability to avoid the obstacle, as Hong et al. [20] reviewed. In Figure 2, there are two different scenarios, in which AGV has to calculate the path using the DWA algorithm. For both scenarios, different weight values will be applied to analyze the consequence of unacceptable configuration. The cases in Figure 2a,c used the same parameters for different scenarios. The same logic is implemented in the cases in Figure 2b,d.

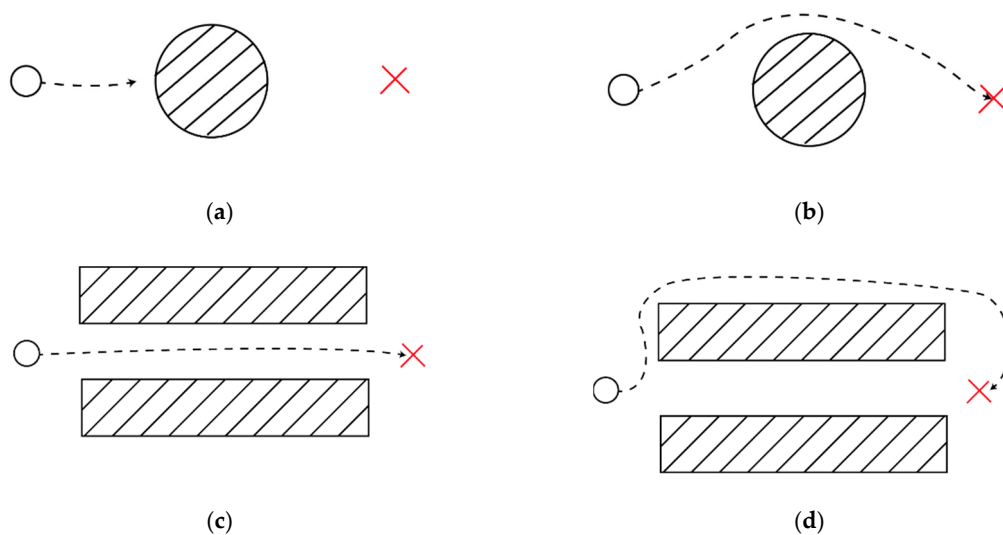


Figure 2. Examples of different dynamic window approach (DWA) situations with different  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$ , as follows: (a,c)  $\alpha_D = 0.8$ ,  $\beta_D = 0.1$  and  $\gamma_D = 0.1$ ; (b,d)  $\alpha_D = 0.1$ ,  $\beta_D = 0.8$  and  $\gamma_D = 0.1$ .

Hence, the ANFIS system will choose these parameters depending on the scenario conditions. For each parameter, an adaptive neuro-fuzzy inference system is proposed. The train will be made individually so as to obtain a better resolution, adopting specific inputs. In the fuzzy DWA, however, the rules are designed manually. This tune is not optimized for the scenario, and depends on human criterion.

With the aim of training ANFIS and simplifying the fuzzy logic controller adjustment, DWA will be executed several times in order to collect data. With these data, ANFIS develops a fuzzy inference system (FIS) using a back-propagation algorithm. Thus, FIS will change the DWA parameters to solve a particular scenario.

This functionality is similar to that of a neural network, in which the inputs go through the membership function and associated parameters. Then, the information will propagate to the output membership function and associated parameters in order to obtain a specific output. Moreover, the associated parameters adapt their values depending on the learning process.

### 3. DWA Improvement

This research will implement two novel concepts for the DWA algorithm. The first one comes from Equation (1), where the classic DWA optimizes the trajectory, just considering the next sample time or step. In this article, however, the DWA algorithm will be executed twice per sample time. In the first execution, DWA analyzed more than one possible path, and in the second one, the DWA analyzed different paths and predicted which one had a better resolution.

The second implementation comes from the DWA constant parameters. As shown in Figure 3, the importance of selecting the correct weights for each situation is essential. To solve this problem, despite using classic fuzzy inference, ANFIS will be implemented.

#### 3.1. Predictive DWA

The classic DWA optimization step will be modified to introduce a prediction step. The optimization function, which it is represented in Equation (1), selects the best path, considering the admissible trajectories. On a predictive DWA (P-DWA), however, it is essential to select more than one path. Otherwise, it cannot implement the prediction step. Hence, for this particular case, DWA calculates the three best trajectories (A, B, and C). These trajectories are considered to be the new possible position for the AGV.

After calculating these three new possible positions, it is time to foresee the future. This step is known as the prediction step, where the DWA will be executed for each best trajectory (A, B, and C), so as to obtain another three paths. Thanks to that, the AGV analyzes nine different positions to arrive. These nine positions are theoretical and are used to make an optimization. Thus, the AGV should select the best path to arrive at one of these nine optimum positions. Nevertheless, the vehicle is just going to move to the A, B, or C position. For example, if the prediction estimated that the fifth position was the best one, the AGV would move to the B position. To have a general idea of this, an example is shown in Figure 3.

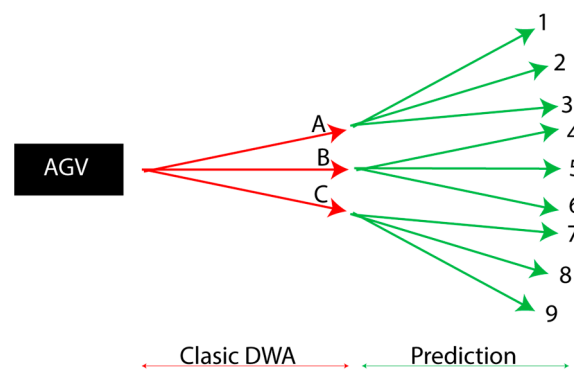


Figure 3. DWA prediction concept. AGV—automated guided vehicle.

Overall, with the normal DWA, the AGV is going to select the best path considering the A, B, or C trajectories. Predictive DWA (P-DWA) gives the chance to analyze the next sample time so as to make a better optimization, as a result of analyzing a large period of time and using more information from the scenario. It is remarkable that it has no sense of how to use a huge prediction time. By using an extensive sample time, the computational cost will be increased considerably. Moreover, by adopting a large period of time, the environment condition could change. Thus, this optimization will not be the best one.

Despite stopping the explanation here, DWA and P-DWA will be compared so as to understand which application has better performance, using the scenarios in Figure 4b,c. The trajectory results are shown in Figure 3, and the analytical results are shown in Tables 3 and 4.

Table 3. DWA and predictive (P)-DWA comparison in train map 2.

Description	DWA	P-DWA
Time to arrive at goal (s)	34.2	32.95
Mean speed (m/s)	0.79	0.82
Minimum distance to obstacle (m)	0.67	0.69
$\alpha_D$	0.5	0.5
$\beta_D$	0.5	0.5
$\gamma_D$	0.5	0.5
$\sigma_D$	3	3
Sample time (s)	0.05	0.05
Prediction Sample time (s)	-	0.5

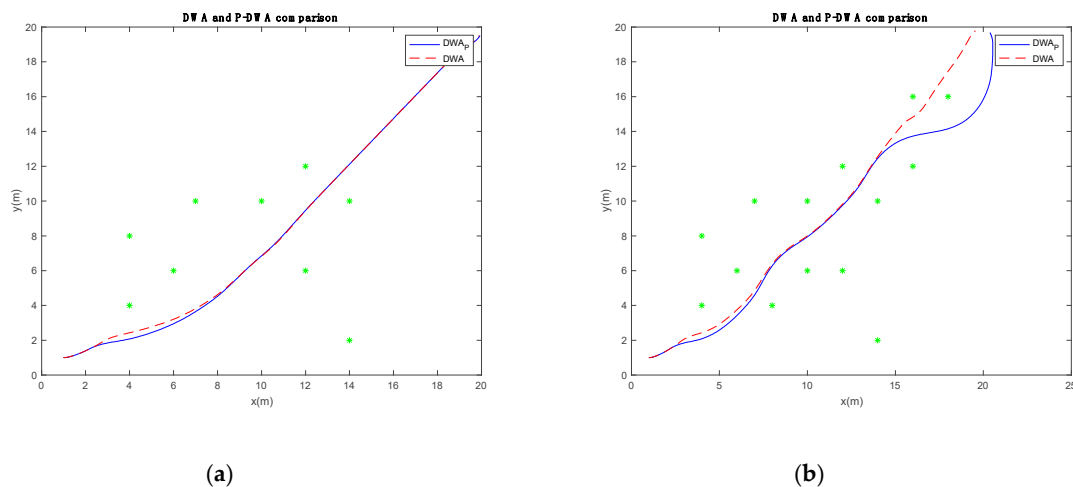


Figure 4. DWA and P-DWA actuation in different scenarios: (a) Map2 and (b) Map3.

In the first execution, for which the trajectory is shown in Figure 4a and the analytics values are represented in Table 3, P-DWA has a better performance, arriving at the goal position in less time. Thus, AGV is faster than the normal DWA. The time difference is about 1.25 s, and the speed value variation is 0.03 m/s.

For this particular case, the P-DWA algorithm is executed every 0.05 s, and it calculates three best paths (A, B, and C), and the prediction window is working with 0.5 s sample time to calculate another three trajectories for each best path.

P-DWA analyzes the future by executing the AGV model, and tries to guess the future by foreseeing what will be happen if it moves to a new point, like whether a model predictive control (MPC) [32] works. Thanks to the prediction window, the algorithm has a better performance and arrives faster at the goal position.

Comparing the paths, both are really similar. P-DWA tries to search an ideal path, giving more space between the AGV and obstacles. With this action, the vehicle can go faster because the  $\text{dist}(V_{agv}, \dot{\gamma})$  calculation has less influence.

The last affirmation can be confirmed with Table 4. For the second comparison, P-DWA diverts more from the last obstacles so as to maintain a top speed. Hence, AGV diverts more from the ideal path, which is a straight line from the start position to goal position. Nevertheless, P-DWA has the capacity to arrive at the goal point faster than DWA, because the classic algorithm has to cross between two obstacles, thus reducing the speed.

**Table 4.** DWA and P-DWA comparison in train map 3.

Description	DWA	P-DWA
Time to arrive at goal (s)	41.25	34.9
Mean speed (m/s)	0.65	0.83
Minimum distance to obstacle (m)	0.44	0.65
$\alpha_D$	0.5	0.5
$\beta_D$	0.5	0.5
$\gamma_D$	0.5	0.5
$\sigma_D$	3	3
Sample time (s)	0.05	0.05
Predictive window(s)	-	0.5

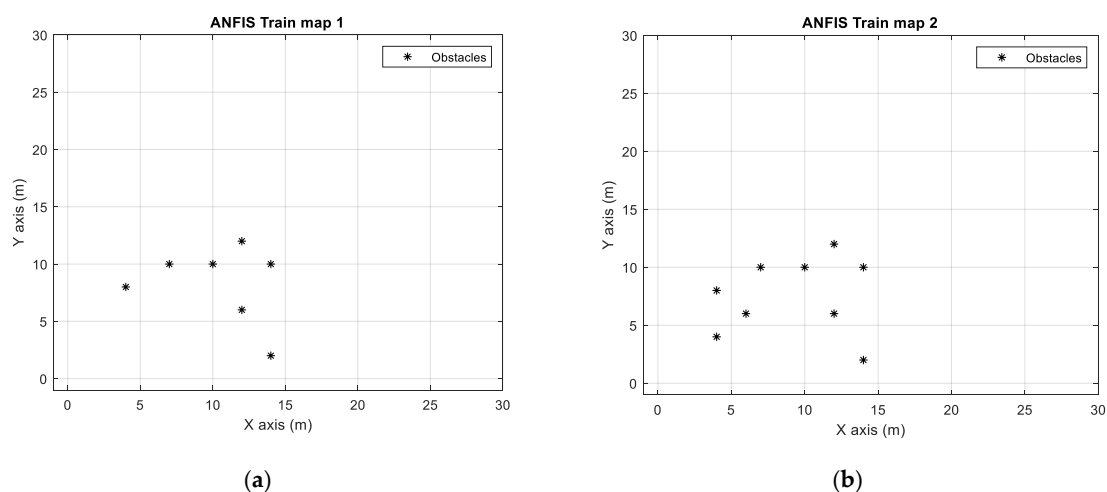
Overall, these comparisons give a general idea about how the DWA path calculation can be optimized, introducing a predictive window. The most relevant parameter is the speed incrementation, where AGV arrives at the goal position in less time. Therefore, there is a path variation between DWA and P-DWA. Depending on the predictive window, the answer will change. Huge predictive windows require a high computational cost. Moreover, if the obstacles change their position, the optimization of the path will not be the best one. Small predictive windows, however, do not affect the path optimization. Thus, P-DWA and DWA will have almost identical paths.

### 3.2. ANFIS Development

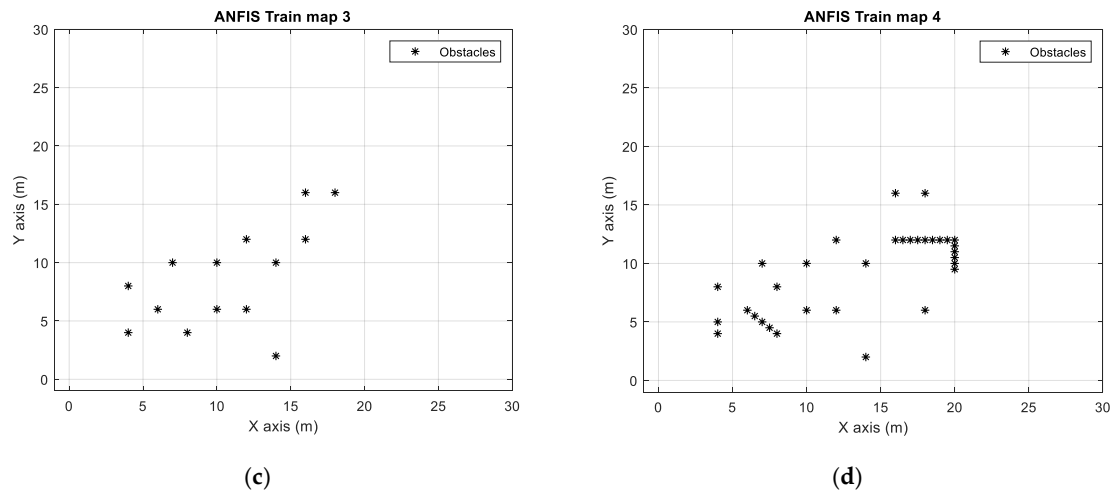
After developing the P-DWA, it is time to evaluate how to adapt the weights using an adaptive neuro-fuzzy inference system. ANFIS, as with the other neural network, depends on the train-data to adjust the fuzzy inference. Therefore, the first action is to develop different scenarios for data acquisition, as shown in Figure 5. In these scenarios, the AGV will move using a classical DWA algorithm, adopting different  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  parameters per execution. It is true that P-DWA has a better performance, as shown in Section 3.1. The purpose of this section; however, it is to obtain the speed, distance to the obstacles, and direction for the target information to train ANFIS.

It has to be considered that when P-DWA uses ANFIS, this fuzzy inference will be executed twice. One for the first step, as with a normal DWA, and another for the prediction step, in order to recalculate the  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  parameters so as to have a better optimization for the trajectory.

The scenarios are built thinking that the AGV starts in pose  $x = 1$ ,  $y = 1$ , and  $\sigma = 1$ , and the goal is in  $x = 20$  and  $y = 20$ . Hence, the obstacles are around the  $y = x$  line, so as to interrupt the AGV best path, which is a straight line between the start point and goal.

**Figure 5.** Cont.





**Figure 5.** Artificial neuro-fuzzy inference system (ANFIS) scenarios for data acquisition, as follows: (a) Map 1 with seven obstacles; (b) map 2 with nine obstacles; (c) map 3 with 14 obstacles; (d) map 4 with 33 obstacles.

The data acquisition is built with 600 samples for each map. In each sample, random  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  values are employed. The random function uses a uniform distribution and the values are from zero to one.

As shown in Figure 2, the DWA behavior depends on the weight values, because these values, for AGV, in some cases, cannot arrive at the goal position. To reject these invalid parameters, a timer will be used. This timer controls how much time the AGV needs to arrive at the goal. Hence, if the AGV does not arrive at the goal position after 250 s, these parameters would be discarded from the data acquisition. Apart from this, in this research, the  $\alpha_D$ ,  $\beta_D$ , and  $\gamma_D$  parameters will use individual ANFIS to optimize the fuzzy inference.

The data acquisition will be transformed to train the data, in order to achieve the ANFIS. Thus, the values have to be prepared using some functions. In Table 5, there are some term definitions that are used in the equations.

**Table 5.** Term definitions.

Term	Definition
h	Heading value for each sample time
H	Mean heading for each data acquisition
v	Speed value for each sample time
V	Mean Speed for each data acquisition
d	Distance to the obstacle value for each sample time
D	Mean Distance for each data acquisition
ArriveGoal	The distance between the AGV and goal for each sample time
$X_{Goal}$	X-position of the goal
$Y_{Goal}$	Y-position of the goal
$X_{AGV}$	X-position of the AGV
$Y_{AGV}$	Y-position of the AGV
n	Number of sample execution
i	Number of sample time

To develop the  $\alpha_D$  fuzzy inference, which is heading to the target correction, ANFIS will be trained using only the headings means, as shown by Equation (2). Apart from this, for this particular case, the n-maximum value will be 2400, considering that there are 600 for each map, and there are four maps.

Moreover, the  $i$ -maximum value must be 5000, as a result of the multiplication of the DWA sample time, which is 0,05 s, and the time limit, which is 250 s.

$$H_n = \frac{\sum h_i}{i} \tag{2}$$

For  $\beta_D$ , which is the object detection parameter, the trained-data will have the AGV mean speed and the mean distance to the obstacle, as in Equation (3) and Equation (4), respectively. In an obstacle avoidance maneuver, it is important to consider the AGV speed, and how far away the obstacle is from the AGV. Nevertheless, if the trained-data only considers the distance to the obstacle, the AGV could avoid some objects with a high speed. In other words, the AGV is going to make an unsafety avoidance.

$$V_n = \frac{\sum v_i}{i} \tag{3}$$

$$D_n = \frac{\sum d_i}{i} \tag{4}$$

Finally, there is  $\gamma_D$ , which controls the speed. The trained data will analyze the mean speed, as shown in Equation (3). The  $\gamma_D$  parameter will have a high value when the AGV is far away from the goal. Nevertheless, when AGV has a top speed, this value has to be low, in order to not influence the obstacle detection and avoidance.

As a future work, the ANFIS trained-data preparation could be optimized. Instead of applying the means of the direction, speed, or distance, the variance of the obstacle to determine a moving obstacle, the divergence from the ideal path for heading parameter, and so on could be adopted. The aim of this research, however, it is to introduce a novel concept of intelligence, which can simplify the fuzzy inference adjustment.

ANFIS, as with other fuzzy inference systems, needs to define the fuzzy type. For this, a Sugeno model was used, as represented in Table 6. Hence, it was used as the default configuration for ANFIS, because the aim of this research is to ensure that this algorithm has the capability to adapt the P-DWA and DWA parameters.

**Table 6.** Configuration of artificial neuro-fuzzy inference system (ANFIS).

Fuzzy Type	Sugeno
And Method	Product of fuzzified input values
Or Method	Probabilistic OR of fuzzified input values
Defuzzy Method	Weighted average of all rule outputs
Implication Method	Scales the consequent membership function by the antecedent result value
Aggregation Method	Sum of the consequent fuzzy sets

The  $\alpha_D$  parameter inference is analyzed in Figure 6. As a consequence of the default configuration, the  $\alpha_D$  membership function has two fuzzy rules, which are optimized to solve four train maps, thanks to the backpropagation algorithm.

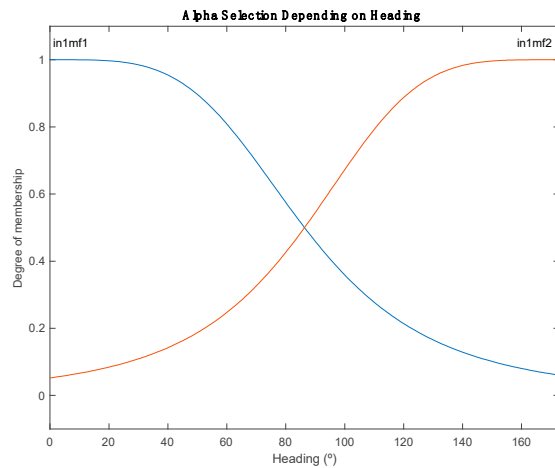


Figure 6.  $\alpha_D$  selection, depending on the AGV heading value.

$\beta_D$  has two inputs, which depend on the obstacle distances and AGV speed. Both inputs are represented in Figure 7. As a result, of adopting two inputs,  $\beta_D$  has two membership functions. Like the other parameter, each membership has two rules.

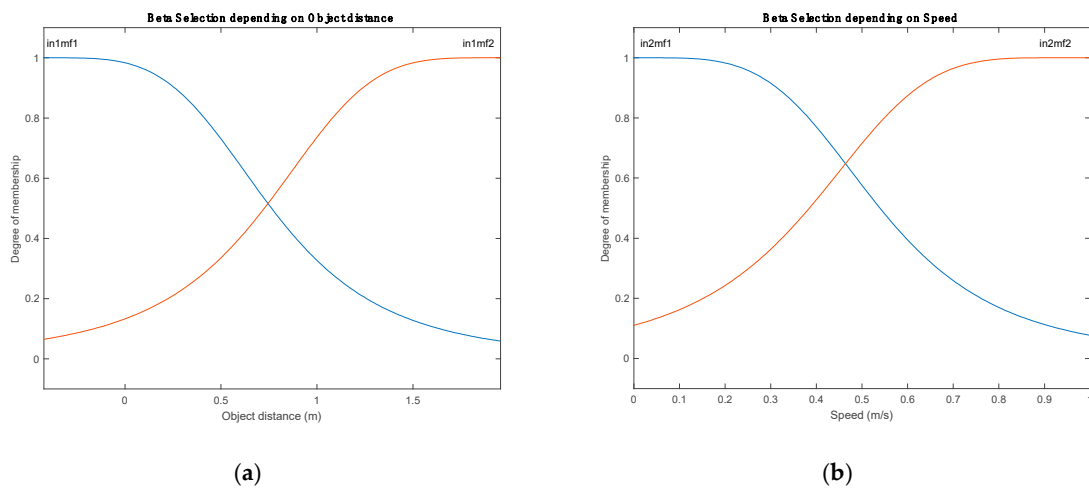


Figure 7.  $\beta_D$  selection depending on two different inputs, namely: (a) Distance to the obstacle and (b) AGV actual speed.

For the last parameter, which is  $\gamma_D$ , the configuration is similar to the first one. It has a single membership function with two rules, as represented in Figure 8. For this time, the membership degree is determined by the mean speeds.

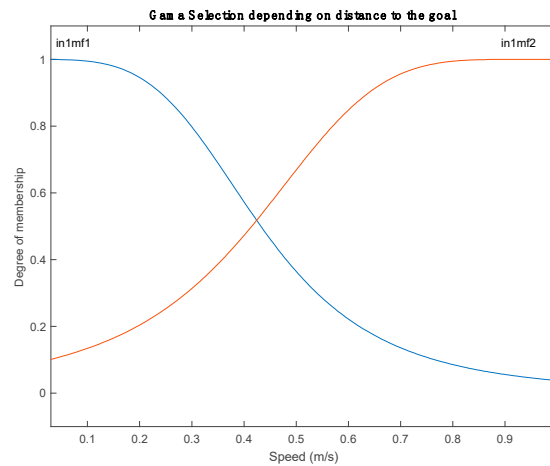


Figure 8.  $\gamma_D$  selection, depending on the distance to the goal position.

#### 4. DWA Improves Results

Now it is time to analyze the AGV behavior, considering the four train-maps. The principal idea is to understand how the parameters change using the ANFIS technique, and to adapt the P-DWA and DWA parameters, depending on the scenario conditions. Moreover, it is remarkable that P-DWA uses the fuzzy inference two times per sample time. In other words, the first time uses FIS to obtain the A, B, and C paths. The second time uses FIS in a prediction step to obtain another nine paths. These results are divided in four sections so as to individually analyze the advantage of this improvement.

##### 4.1. Map 1 Results

Map 1 corresponds to Figure 5a. In Table 7 and Figure 9 are the values for comparing DWA ANFIS with P-DWA ANFIS.

Table 7. Map 1 data comparison.

Description	P-DWA ANFIS	DWA ANFIS
Time to arrive at goal (s)	31.55	33.95
Mean speed (m/s)	0.86	0.8
Mean heading ( $^{\circ}$ )	151.55	150.96
Minimum distance to obstacle (m)	0.69	0.74

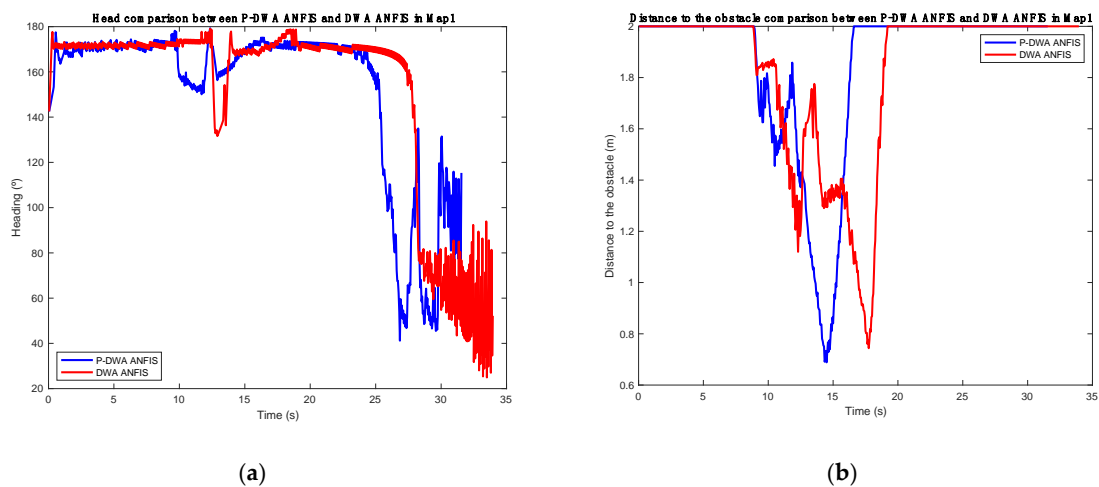
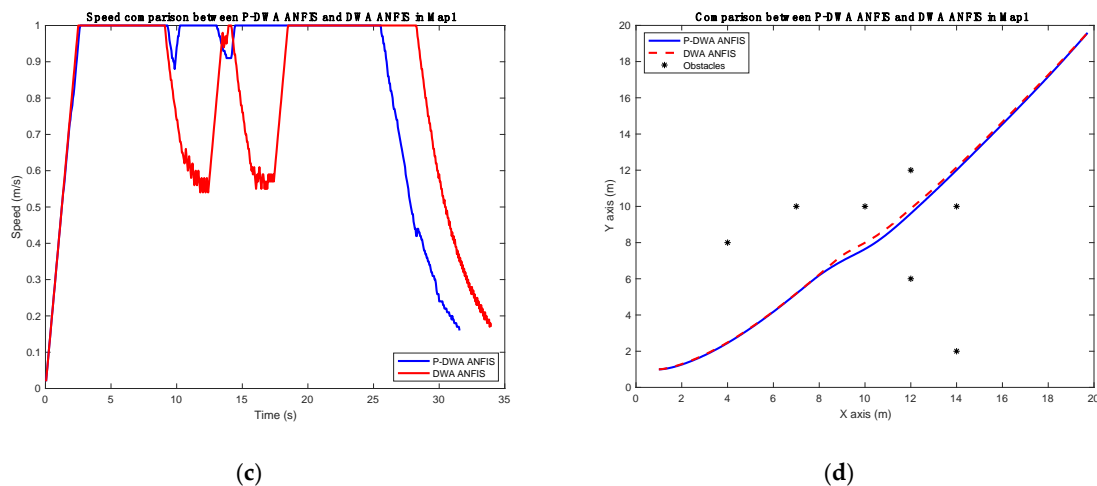


Figure 9. Cont.



**Figure 9.** P-DWA ANFINS and DWA ANFIS values comparison during the execution, as follows: (a) Heading comparison, (b) distance to the obstacle comparison, (c) speed comparison, and (d) AGV trajectories with different algorithms.

As the results show, P-DWA ANFIS is faster than DWA ANFIS, because the prediction step can analyze the future and selects the path with the least speed variation. This affirmation can be confirmed with Figure 9c. DWA needs to reduce the speed from 10 s to 13 s, as a result of not considering obstacle  $x = 10$  and  $y = 10$  in the trajectory. Moreover, in the same period of time, the direction to the target has a value change, caused by the obstacle avoidance maneuver. DWA ANFIS, however, is more conservative in obstacle avoidance, hence, it maintains more distance between the obstacle and AGV. Apart from these observations, the trajectory of both functions is almost identical in first ten meters and in the last six meters. Thus, ANFIS works similarly for both algorithms, and the difference comes from the prediction that anticipates to the approaching obstacle and modifies the path searching for a new space, reducing the speed as little as possible.

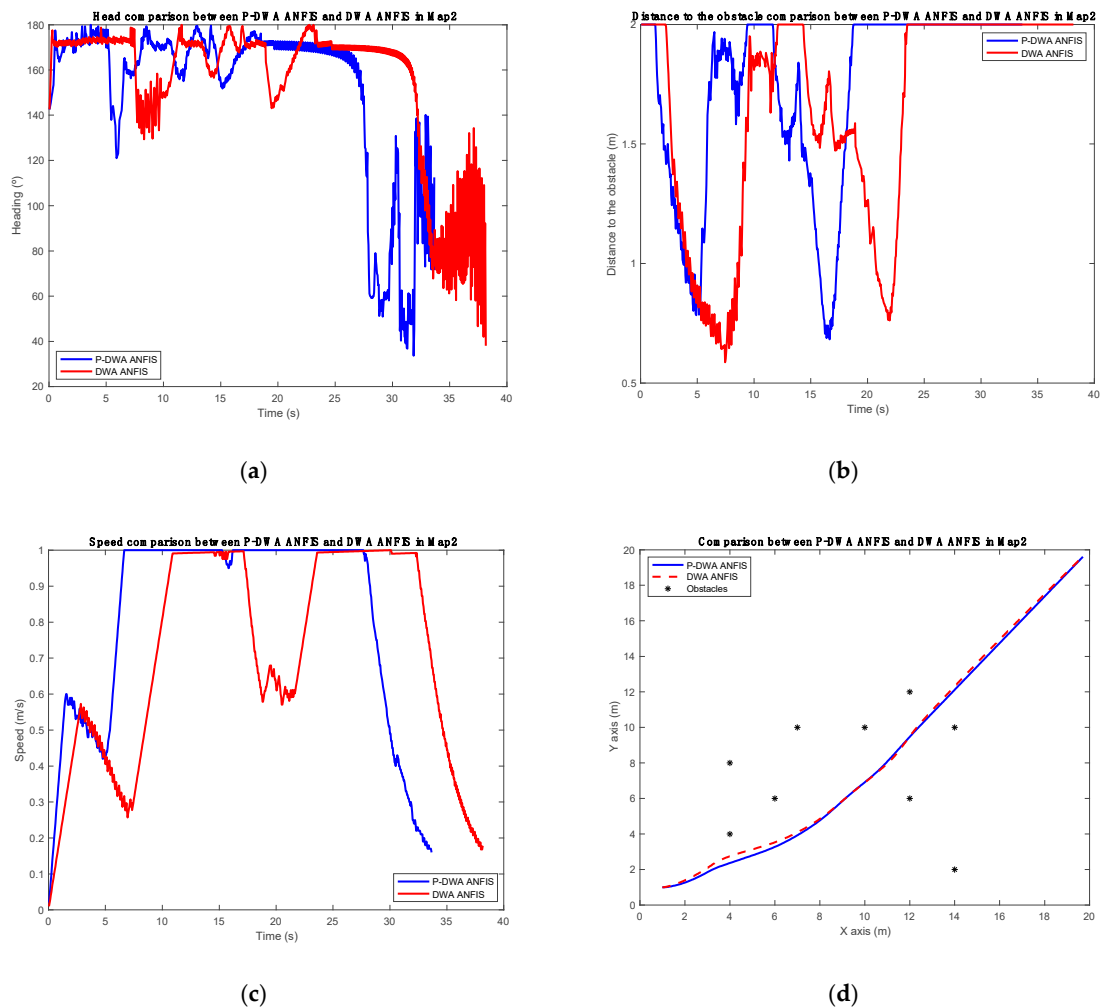
Furthermore, Figure 9a–c appreciate some AGV model aspects. For example, at the maximum top speed, which is 1 m/s, AGV will be aligned with the goal position when the heading value is  $180^\circ$  and AGV has a 2 m visibility for the obstacle.

#### 4.2. Map 2

Adopting the same evaluation method as in Section 4.1, map 2 analytics results are represented in Table 8 and Figure 10.

**Table 8.** Map 2 data comparison.

Description	P-DWA ANFIS	DWA ANFIS
Time to arrive at goal (s)	33.65	38.15
Mean speed (m/s)	0.81	0.71
Mean heading ( $^\circ$ )	151.31	154.35
Minimum distance to obstacle (m)	0.68	0.56



**Figure 10.** DWA ANFINS parameters change during the scenarios, as follows: (a)  $\alpha_D$  change during the execution, (b)  $\beta_D$  change during the execution, (c)  $\gamma_D$  change during the execution, and (d) AGV trajectories with different configurations.

For map 2, as with the map 1 results, P-DWA ANFIS is faster than DWA ANFIS. The predictive algorithm needs less time to arrive at the top speed, because it starts considering the  $x = 4$  and  $y = 4$  obstacle earlier, as shown in Figure 10b. Apart from that, DWA takes another route, which is closer to the obstacle so as to avoid the object, causing a huge speed reduction. Furthermore, if Table 3 is considered, it seems that ANFIS has the worst performance. For this time and with this particular configuration, the classic DWA or P-DWA has better time results. This problem can be solved, however, by applying another time restriction for data-trained rejection. By modifying this restriction, which is 250 s, ANFIS will have another performance, because the trained-data has faster solutions.

According to the map 1 conclusions, for this execution, the global idea maintains. In other words, the P-DWA ANFIS continues to have a better performance than DWA ANFIS. Moreover, the trajectory continues, being almost identical in the beginning and the end. Nevertheless, there is a difference this time. The DWA ANFIS gives less distance to the obstacle. This is the conclusion of Section 3.1. In Section 4.1, however, this the global idea changes.

#### 4.3. Map 3

As with the rest of the subsections, the map 3 results are represented in Table 9 and Figure 11.

Table 9. Map 3 data comparison.

Description	P-DWA ANFIS	DWA ANFIS
Time to arrive at goal (s)	46.7	50.7
Mean speed (m/s)	0.59	0.54
Mean heading (°)	153.47	156.20
Minimum distance to obstacle (m)	0.47	0.46

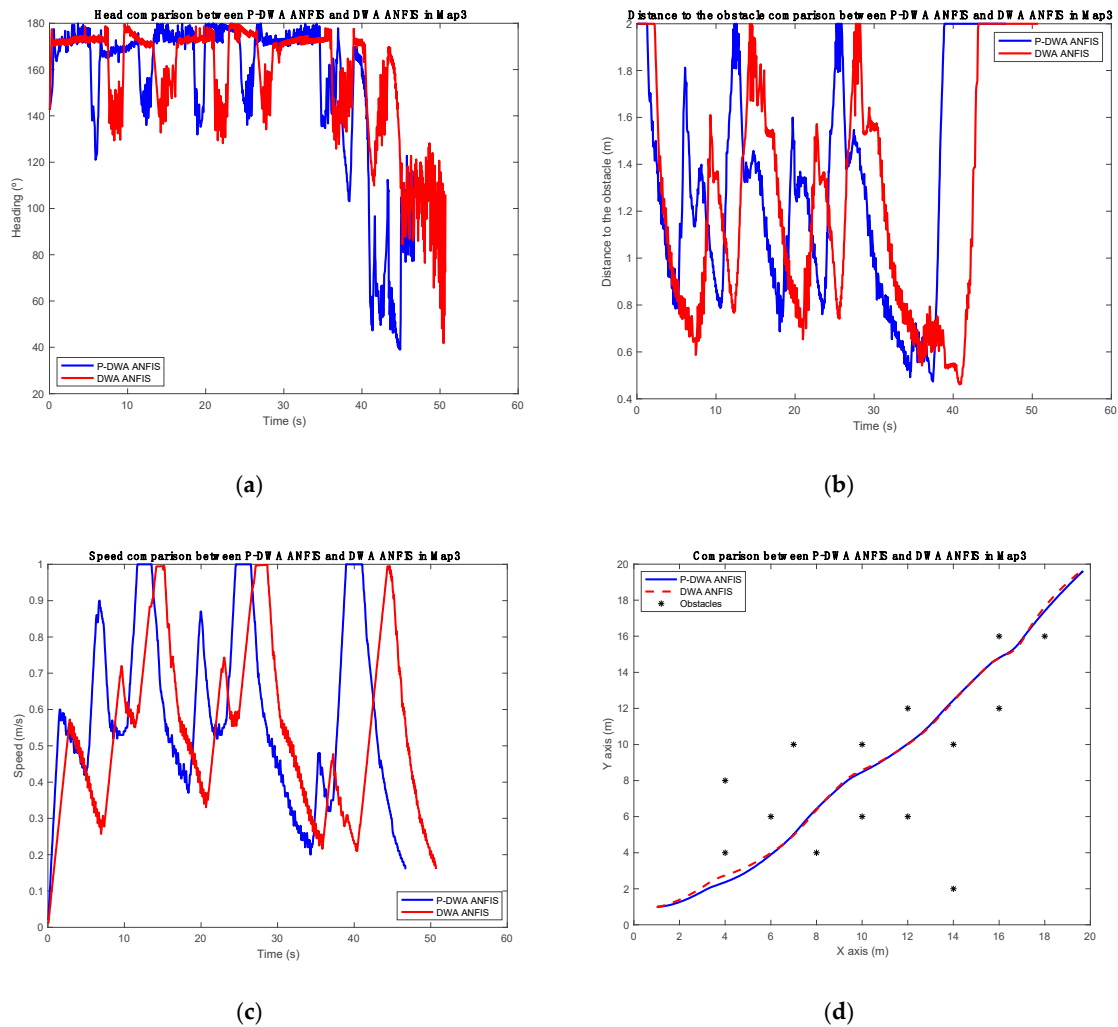


Figure 11. DWA ANFINS parameters change during the scenarios, as follows: (a)  $\alpha_D$  change during the execution, (b)  $\beta_D$  change during the execution, (c)  $\gamma_D$  change during the execution, and (d) the AGV trajectories with different configurations.

To not repeat the same explanation, the general ideas will be mentioned in Sections 4.3 and 4.4. For Section 4.3, the global idea maintains. Overall, the P-DWA ANFIS is faster than DWA ANFIS, thanks to the prediction step. Moreover, the ANFIS trained data can be improved using better resolution samples, because Table 4 shows a better performance than Table 9. Apart from that the P-DWA is more conservative, maintaining more space between the AGV and obstacles. Therefore, the idea is that in general, P-DWA safely gains strength.

#### 4.4. Map 4

Adopting the same logic from other cases, map 4 results are represented in Table 10 and Figure 12.

Table 10. Map 4 data comparison.

Description	P-DWA ANFIS	DWA ANFIS
Time to arrive at goal (s)	57.75	67.60
Mean speed (m/s)	0.49	0.41
Mean heading (°)	153.7	159.66
Minimum distance to obstacle (m)	0.2	0.08

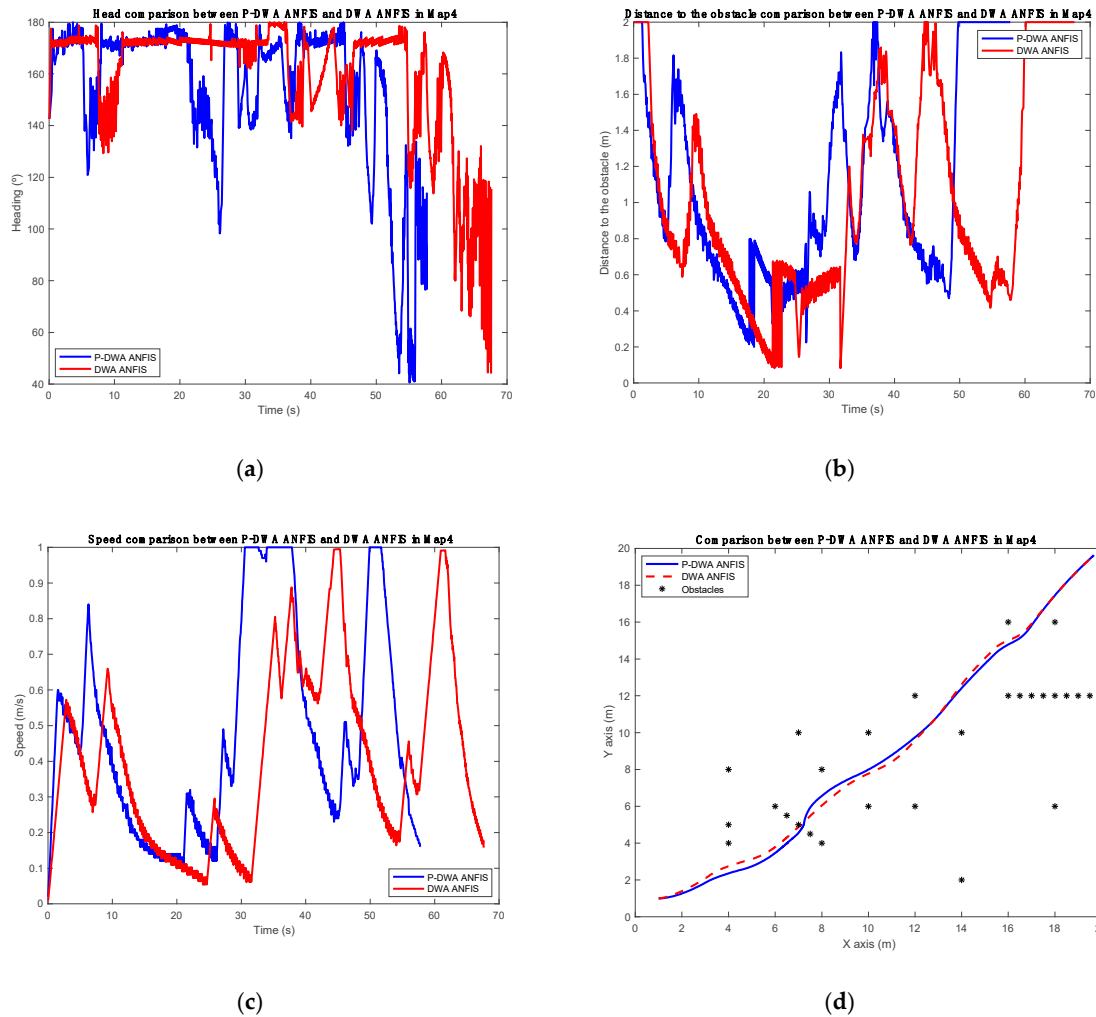


Figure 12. DWA ANFINS parameters change during the scenarios, as follows: (a)  $\alpha_D$  change during the execution, (b)  $\beta_D$  change during the execution, (c)  $\gamma_D$  change during the execution, and (d) AGV trajectories with different configurations.

This is the most chaotic case. In Figure 12d, AGV crosses between the obstacles, reducing the speed significantly. Hence, AGV needs more time to arrive at the goal position. To improve these results, the ANFIS train data could be more restrictive. Thus, in the data selection process, more restriction could be applied to avoid these situations, such as, AGV cannot cross between two obstacles. Nevertheless, as Figure 12b shows, AGV continues to have enough space to cross it. The minimum distance to the obstacle for DWA ANFIS is 0.08 m, which is really close. Hence, in comparison with P-DWA ANFIS, this action significantly reduces the speed of the vehicle, arriving more than 10 s later to the goal position.

For this resolution, different to the other tested maps, the paths are almost identical in the end. Furthermore, P-DWA ANFIS tries to predict an ideal path to solve the first obstacles. The prediction step, however, is not large enough to solve this situation, and the ideal path crosses between the obstacles.



In other words, this ANFIS optimization should be made individually for each map. Thus, the AGV could improve the resolution, because the weight calculations are trained for a particular scenario. Moreover, the restriction in trained-data should give a better resolution for this chaotic case. Apart from that, in these situations, offline path planning could be used.

## 5. Conclusions

The prediction step and ANFIS technique improvements were proposed in this research work. The development of a P-DWA increases the path mean speed, because the prediction step has better optimization than the DWA. In other words, the prediction step can analyze what will be happen in the next steps, and it can anticipate some situations. Thus, it could be possible to obtain a better performance for the path.

In comparison, P-DWA is faster than DWA, adopting an alternative trajectory to maintain the top speed, despite reducing speed and crossing between two obstacles. Thus, P-DWA maintains more distance between the AGV and obstacle, doing a safety avoidance maneuver.

Furthermore, the prediction sample time has a fixed value, which is 0.5 s. It could be interesting to analyze the behavior of the P-DWA algorithm with different values. Obviously, the low or high values can alter the improvement of the prediction step. With large prediction values, the dynamic obstacles can change the position, hence, the prediction would be wrong. With low values, the algorithm does not have enough information to find an optimum path to avoid an obstacle. Therefore, DWA and P-DWA would have the same behavior.

Apart from P-DWA, the ANFIS algorithm is introduced to optimize the fuzzy inferences. The results show that ANFIS can improve the P-DWA and DWA behavior, changing the parameters depending on the scenario conditions. This configuration is optimized to solve four maps. To improve the ANFIS performance, this algorithm should have a particular training for each map.

Moreover, it is remarkable that the trained-data selection is not the best one to easily solve all of the maps. Nevertheless, adopting different restrictions, such as not crossing obstacles, if the gap is less than a particular distance; speed restriction; and time restriction, the ANFIS behavior can be improved significantly. The aim of this research is not to obtain the best fuzzy inference to solve these four maps. It is to introduce a novel system to develop a classic motion controller, which uses static parameters.

For future work, it could be interesting to analyze the obstacles variance to determinate if there is any moving obstacle. Considering this parameter, AGV has to change speed and direction in order to solve this particular case. Furthermore, it could be interesting to adopt different restrictions to improve the ANFIS resolution, like implementing a more restrictive rejection time.

**Author Contributions:** D.T.-F.-B., E.Z., and U.F.-G. developed and programmed the simulation set up. They also wrote the manuscript. A.S.-A. and R.M. made constructive contributions in the process of preparing the paper.

**Funding:** This research was financed by the plant of Mercedes-Benz Vitoria through PIF program to develop an intelligent production.

**Acknowledgments:** The authors are grateful to the Government of the Basque Country and to the University of the Basque Country UPV/EHU through the SAIOTEK (S-PE11UN112) and EHU12/26 research programs, respectively. The plant of Mercedes-Benz Vitoria is also acknowledged in especially to Emilio Titos (the managing director of the Mercedes-Benz Vitoria plant), Jose Carlos Velasco (director of the Mercedes-Benz Vitoria final assembly plant), the final assembly maintenance department of Mercedes-Benz Vitoria, Javier Loredó (director of final assembly maintenance department), Javier Gómez, Jose Antonio Hernando and Tomás Hernandez to give the opportunity to makes this research in intelligent production.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jin, X.; Su, T.; Kong, J.; Bai, Y.; Miao, B.; Dou, C. State-of-the-Art Mobile Intelligence: Enabling Robots to Move Like Humans by Estimating Mobility with Artificial Intelligence. *Appl. Sci.* **2018**, *8*, 379. [[CrossRef](#)]

2. Ortigoza, R.S.; Marcelino-Aranda, M.; Ortigoza, G.S.; Guzman, V.M.H.; Molina-Vilchis, M.H.; Saldana-Gonzalez, G.; Herrera-Lozada, J.C.; Olguin-Carbajal, M. Wheeled Mobile Robots: A review. *IEEE Lat. Am. Trans.* **2012**, *10*, 2209–2217. [[CrossRef](#)]
3. Bonin-Font, F.; Ortiz, A.; Oliver, G. Visual Navigation for Mobile Robots: A Survey. *J. Intell. Robot. Syst.* **2008**, *53*, 263–296. [[CrossRef](#)]
4. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [[CrossRef](#)]
5. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [[CrossRef](#)]
6. Buniyamin, N.; Wan Ngah, W.A.j.; Mohamad, Z. PointsBug versus TangentBug algorithm, a performance comparison in unknown static environment. In Proceedings of the 2014 IEEE Sensors Applications Symposium (SAS), Queenstown, New Zealand, 18–20 February 2014; pp. 278–282.
7. Ever, Y.K. Using simplified swarm optimization on path planning for intelligent mobile robot. *Proced. Comput. Sci.* **2017**, *120*, 83–90. [[CrossRef](#)]
8. Mei, Z.; Chen, Y.; Jiang, M.; Wu, H.; Cheng, L. Mobile robots path planning based on dynamic movement primitives library. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 6906–6911.
9. Teso-Fz-Betoño, D.; Zulueta, E.; Fernandez-Gamiz, U.; Aramendia, I.; Uriarte, I. A Free Navigation of an AGV to a Non-Static Target with Obstacle Avoidance. *Electronics* **2019**, *8*, 159. [[CrossRef](#)]
10. Lopez, F.G.; Abbenseth, J.; Henkel, C.; Dörr, S. A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications. In Proceedings of the 2017 European Conference on Mobile Robots (ECMR), Paris, France, 6–8 September 2017.
11. Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore, 29 May–3 June 2017.
12. Yu, H.; Duan, J.; Taheri, S.; Cheng, H.; Qi, Z. A model predictive control approach combined unscented Kalman filter vehicle state estimation in intelligent vehicle trajectory tracking. *Adv. Mech. Eng.* **2015**, *7*, 168781401557836. [[CrossRef](#)]
13. Qingjia, C.; Rongjun, D.; Bing, Z.; Xiaojian, W. Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering. *J. Autom. Eng.* **2018**, *232*, 1237–1252.
14. Sathiya, V.; Chinnadurai, M. Evolutionary Algorithms-Based Multi-Objective Optimal Mobile Robot Trajectory Planning. *Robotica* **2019**, *37*, 1363–1382. [[CrossRef](#)]
15. Fan, X.; Luo, X.; Yi, S.; Yang, S.; Zhang, H. Optimal path planning for mobile robots based on intensified ant colony optimization algorithm. In Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003, Changsha, Hunan, China, 8–13 October 2003.
16. Dai, X.; Long, S.; Zhang, Z.; Gong, D. Mobile Robot Path Planning Based on Ant Colony Algorithm With A Heuristic Method. *Front. Neurobotics* **2019**, *13*, 15. [[CrossRef](#)] [[PubMed](#)]
17. Yang, H.; Qi, J.; Miao, Y.; Sun, H.; Li, J. A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization. *IEEE Trans. Indust. Electron.* **2019**, *66*, 8557–8566. [[CrossRef](#)]
18. Bellotto, N.; Hu, H. Multisensor-Based Human Detection and Tracking for Mobile Service Robots. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* **2009**, *39*, 167–181. [[CrossRef](#)] [[PubMed](#)]
19. Truong, X.; Yoong, V.N.; Ngo, T. Dynamic social zone for human safety in human-robot shared workspaces. In Proceedings of the 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Kuala Lumpur, Malaysia, 12–15 November 2014.
20. Hong, Z.; Chun-Long, S.; Zi-Jun, Z.; Wei, A.; De-Qiang, Z.; Jing-Jing, W. A Modified Dynamic Window Approach to Obstacle Avoidance Combined with Fuzzy Logic. In Proceedings of the 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Guiyang, China, 18–24 August 2015.
21. Abubakr, O.A.; Jaradat, M.A.K.; Hafez, M.A. A reduced cascaded fuzzy logic controller for dynamic window weights optimization. In Proceedings of the 2018 11th International Symposium on Mechatronics and its Applications (ISMA), Sharjah, United Arab Emirates, 4–6 March 2018.

22. Arras, K.O.; Persson, J.; Tomatis, N.; Siegwart, R. Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002.
23. Salleh, M.A.M.; Talpur, N.; Hussain, K. Adaptive Neuro-Fuzzy Inference System: Overview, Strengths, Limitations, and Solutions. In *Lecture Notes in Computer Science*; Springer: Cham, Germany, 2017.
24. Jang, J.-S.R. ANFIS: Adaptive-Network-based Fuzzy Inference System. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]
25. Zamani, H.A.; Rafiee-Taghanaki, S.; Karimi, M.; Arabloo, M.; Dadashi, A. Implementing ANFIS for prediction of reservoir oil solution gas-oil ratio. *J. Nat. Gas Sci. Eng.* **2015**, *25*, 325–334. [[CrossRef](#)]
26. Vakhshouri, B.; Nejadi, S. Comparison of Fuzzy Logic and ANFIS for Prediction of Compressive Strength of RMC. *Neurocomputing* **2018**, *280*, 13–22. [[CrossRef](#)]
27. Lu, J.; Kintak, U. Mobile robot navigation based on adaptive neuro-fuzzy inference system with virtual target strategy. In Proceedings of the 2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Ningbo, China, 9–12 July 2017.
28. Singh, M.K.; Parhi, D.R.; Pothal, J.K. ANFIS Approach for Navigation of Mobile Robots. In Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kerala, India, 27–28 October 2009.
29. Imen, M.; Mansouri, M.; Shoorehdeli, M.A. Tracking control of mobile robot using ANFIS. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August 2011.
30. Pandey, A.; Kumar, S.; Pandey, K.K.; Parhi, D.R. Mobile robot navigation in unknown static environments using ANFIS controller. *Perspec. Sci.* **2016**, *8*, 421–423. [[CrossRef](#)]
31. Kim, Y.N.; Ko, D.W.; Suh, I.H. Confidence random tree-based algorithm for mobile robot path planning considering the path length and safety. *Int. J. Advan. Robot. Syst.* **2019**, *16*, 172988141983817. [[CrossRef](#)]
32. Žáčková, E.; Přívara, S.; Váňa, Z.; Cigler, J. Building modeling: On selection of the model complexity for predictive control. *Comput. Aided Chem. Eng.* **2013**, *32*, 205–210. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).