ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN INTELLIGENT NEURAL ROBOTICS FPGA DESIGN FOR PATH IDENTIFICATION

VISHAL ANDRA[1], S. ROOBAN[2]

[1]M. Tech student, Robotics and Automation, ECE Department, KLEF Deemed to be University, Green Fields, Vaddeswaram, Guntur District, A.P., INDIA. 2201170001ECE@gmail.com
[2]Associate Professor, ECE Department, KLEF Deemed to be University, Green Fields, Vaddeswaram, Guntur District, A.P., INDIA.

ABSTRACT

Autonomous robots' ability to navigate various dynamic situations effectively depends on their ability to plan their paths and avoid obstacles. There are many approaches to path planning and Obstacle avoidance, and it attained an aimless way. A path without any preplanned route is called aimless. To reach its destination, the robot keeps moving randomly in one direction. Paths without goals are risky and inefficient. In addition to possibly travelling farther than is necessary, the robot might run into roadblocks. Therefore, a novel Decision Zfnet Path Planning System (DZPPS) was developed in this work. Initially, the robot's behaviour was trained using the sensed data. Then, the path was identified for the robot. After that, the robot was supposed to avoid obstacles by moving away from them when they were encountered with the assistance of the planning system. The decision function is used to avoid obstacles while driving the robot. In conclusion, measures such as path length, path efficiency, planning time, and distance to obstacles were used to determine the effectiveness of the suggested planning system.
Keywords: *Field Programmable Gate Array, Path Planning, Obstacle Avoidance, Flipflop, Lookup Table*

1. INTRODUCTION

In the last few years, robot development, which can help people with their daily duties, has gained popularity as a field of study [1]. Providing different services at shopping malls, entertaining guests at exhibitions and function halls, and helping the elderly and physically disabled in their homes are potential application situations for these service robots [2, 3]. A mobile robot's primary function is path planning; this allows it to navigate its surroundings using sensory data [4]. Computational vision has excellent promise for robotic navigation, and in the past ten years, there has been a lot of research done on vision-based path identification [5]. Path identification aims to travel from one place to another without incident or obstacles [6]. One of the most crucial assignments for mobile robots is to determine the shortest or best path [7]. In robotics, the best course may be minimizing the number of turns, computation time, and distance from source to destination [8]. Planning the course the robot needs to take is part of this navigation control system [9]. When the start and goal positions are known about the same reference system in the robot's navigation environment, the path planning system calculates the robot's current position along the path [10]. The depiction of the environment can be discrete or continuous, static or dynamic [11]. Avoiding collisions with obstacles and giving the shortest path or time from the start to the goal point is essential for optimizing path planning. There are various path-planning implementations for this [12, 13].Much emphasis has been paid to neural network techniques for creating real-time robot paths [14]. Path identification for mobile robots using a probabilistic learning technique: this approach computes possible robot paths using a local algorithm, query, and learning phases [15]. A neural network model for navigation mobile robot techniques uses supervised learning to create dynamic pathways that avoid barriers [16]. However, this approach is computationally complex because it combines the effector control transform model with direction-to-rotate with the vector associative map model [17, 18]. Additionally, given that the model produces distinct paths depending on specific parameter values, it is unclear whether the given way is optimal [19]. In certain instances, the aimless path search may also provide results [20]. To solve these issues, this paper proposed a novel methodology based on the decision Zfnet system for clear path identification. The primary contribution of the current research project is described as follows:

- Initially, the sensed data was gathered and utilized to train the robot's behaviour
- Then, a novel DZPPS was designed to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

determine the ideal path for the robot's navigation
- Next, the robot ought to avoid obstacles when they are encountered with the help of the planning system
- Finally, the effectiveness of the suggested planning system was calculated by metrics like path length, path efficiency, planning time, and distance to obstacles.

The paper is given in section two as the recent study, and the solution for the specified problem is further developed in section three. The validated results of the novel solution are discussed in section four. The research work was concluded in section five.

2. RELATED WORK

A few recent related works are explained as follows, Sharma et al. [21] suggested a novel method that combines a path-planning algorithm with a snake algorithm. The proposed approach improves robot performance for a range of uses. The integrated algorithm efficiently detects obstacles and their borders. This improves robots' ability to plan courses free of collisions. Positive outcomes imply that the suggested strategy has numerous uses in robot path planning. However, for multid's computation, time is not optimized.

The multi-robot system algorithm goes beyond traditional Q-learning's drawbacks, namely its sluggish learning speed and time consumption, by combining Q-learning with the APF approach Pantrigo et al. [22]. The model's findings show how the QAPF algorithm can improve path planning efficiency and learning speed. In every test situation, the QAPF learning algorithm performs better than traditional Q-learning. It is more dependable for mobile robot path planning because it needs fewer episodes to provide better path-planning outcomes and has less variability in training time. However, a variety of path-planning issues cannot be resolved with this method.

Wang et al. [23] present a brand new method called GA-RRT that blends an A* cost function with goal probability bias. The efficiency and search time of the conventional RTL algorithm are greatly enhanced by the GA-RRT method. Locally optimized pathways are also produced by it. The GA-RRT algorithm is a viable dual-arm robot cooperative obstacle avoidance path identification method. However, the search efficiency of the algorithm was poor.

The modified A* algorithm presented in this research addresses the problems of numerous turning points and acute angles in the search path Qian et al. [24]. The suggested technique uses a cubic Bezier curve and an adaptive step size adjustment strategy to optimize the way for a shorter runtime and more efficient robot motion. Planning the best route involves merging information linked to the global path to optimize safety performance, smoothness, and route length. However, the robotic path planning algorithms do not apply to multiple situations and tasks.

The presented method efficiently generates collision-free pathways in familiar environments using the PRM algorithm Alarabi et al. [25]. It performs better regarding turns and path distance than approaches based on genetic algorithms. When determining optimal pathways, the PRM algorithm outperforms another method in terms of efficiency. However, this method might not work well in dynamic or unpredictable situations because it is restricted to known surroundings.

2.1 Problem Statement:

Designing an intelligent neural robotics system with Field-Programmable Gate Array (FPGA) technology for path identification involves several key challenges. One of the primary challenges is to develop a neural network-based algorithm that can accurately identify paths or routes in a given environment. This algorithm needs to be trained on a diverse set of data to ensure robustness and adaptability to different environments and conditions.

3. PROPOSED METHODOLOGY

A novel Decision Zfnet Path Planning System (DZPPS) is a state-of-the-art method that combines FPGA (Field-Programmable Gate Array) hardware to produce an intelligent robotic path identification solution. Initially, the robot's behaviour was trained using the detected data. Then, the proposed planning system identifies the optimal path for the robot to follow and avoids the obstacles by moving away from them when they are encountered. Finally, the system's performance was measured with the following metrics: path length, path efficiency, planning time, and distance to obstacles.

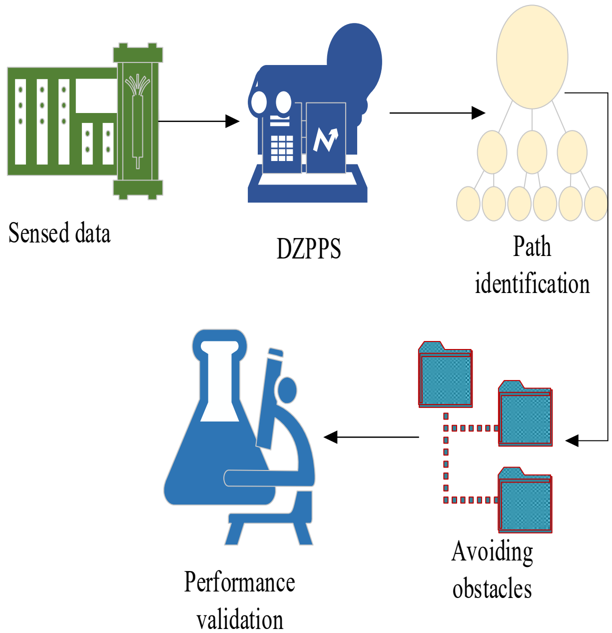

Fig.1: Proposed Architecture

The suggested architecture is displayed in Fig. 1. The proposed method is based on the combined decision tree process [30] and Zfnet [31]. The process can be done using the layers of the Zfnet. The zfnet contains the input, convolutional, pooling, and output layers. The processing can be done through the convolutional layer.

3.1 Process of the proposed DZPPS

The processing layers of the DZPPS method are described in Fig.2. The dataset was gathered and trained in the input layer. Then, path identification and obstacle avoidance can be done through the convolutional layer. Finally, the path outcome was brought into the output layer. The process of the DZPPS methods begins with data training. Here, the sensed data was collected and trained in the system.

$$T(S) = Data(S) * \sum_{i=1}^{p} (P_i)^2 \quad (1)$$

The data training can be done using Eqn. (1). Where S represents the sensed data, T the training function variable, p the number of pathways, and P_i the probability of an instance S belonging to the class i.

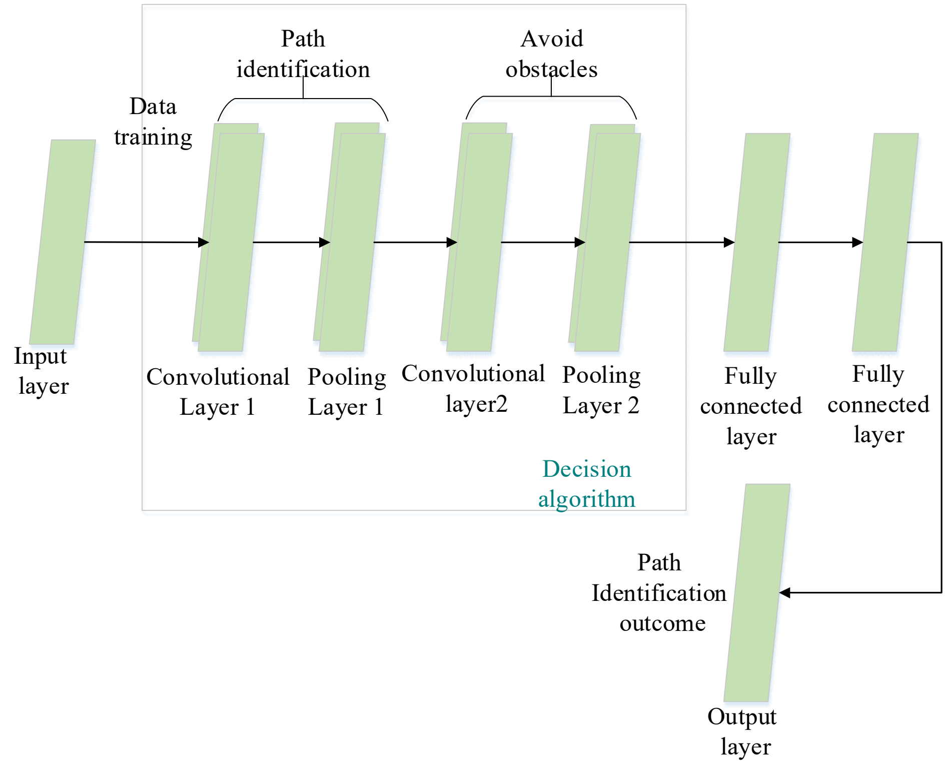

Fig.2: The Processing Layers Of The Proposed DZPPS

3.1.1 Path Identification

Once the dataset is trained to the model, it applies to path planning. When the robot must navigate an environment, the decision tree selects the best path to follow, considering the current state and the desired goal. The path identification can be done using Eqn. (2).

$$PI = Data(S) + \alpha(S) \times \frac{D_e}{T_e} \tag{2}$$

Where PI represents the path identification result variable, α describes the decision variable used to process the sensed data, D_e indicates the distance of the destination, which is the distance of the robot's current position to its destination and T_e represents the time to reach the goal. Based on this equation, the path identification process can be done.

3.1.2 Obstacle avoiding

An essential component of autonomous robotics is obstacle avoidance, which allows robots to go through challenging environments safely and effectively. It entails planning a collision-free route to the goal while simultaneously identifying and avoiding impediments. The obstacle-avoiding process can be done using Eqn. (3).

$$OA = \begin{cases} if\,(D<S \;\; and \; L) & Left \\ if\,(D<S \;\; and \; \neg L \; and \; R) & Right \\ if\,(D<S \; and \; \neg L \; and \neg R) & Stop \\ if\,(D \geq S) & Forward \end{cases} \tag{3}$$

Where, OA represents the Obstacle-avoiding variable, D the distance to the Obstacle, S the safe distance, L the clear path to the left, and R the clear direction to the right. The decision function determines how moving away the closest Obstacle is. It decides whether it needs to halt or move left or right if the distance is less than a safe threshold. It advances if there is a safe distance between it and the obstruction.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm.1: DZPPS

```
Start
{
        Data training()
        {
                int S, p, P_i, T;
                //Initializing the data training variable
                Train → Data(S) * path
                // Trained the sensed data to the robot
        }
        Path identification()
        {
                int PI, D_e, T_e, α;
                // Initialized the path identification variables
                Path → Data(S) + α(Distance / Time)
                // identified the path
        }
        Obstacle avoiding()
        {
                if(D < S, L)
                {
                Move Left
                }
                if(D < S, ¬L, R)
                {
                Move Right
                }
                if(D < S, ¬L, ¬R)
                {
                Stop
                }
                if(D ≥ S)
                {
                Move forward
                }
                // Avoiding obstacles
        }
}
Stop
```

The complete explanation of the steps and processes included in the suggested model may be found in Algorithm.1. These detailed instructions were followed while running the Python code, and the results were verified. Using the pseudocode style, all parameters of mathematical functions were integrated into the algorithm.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

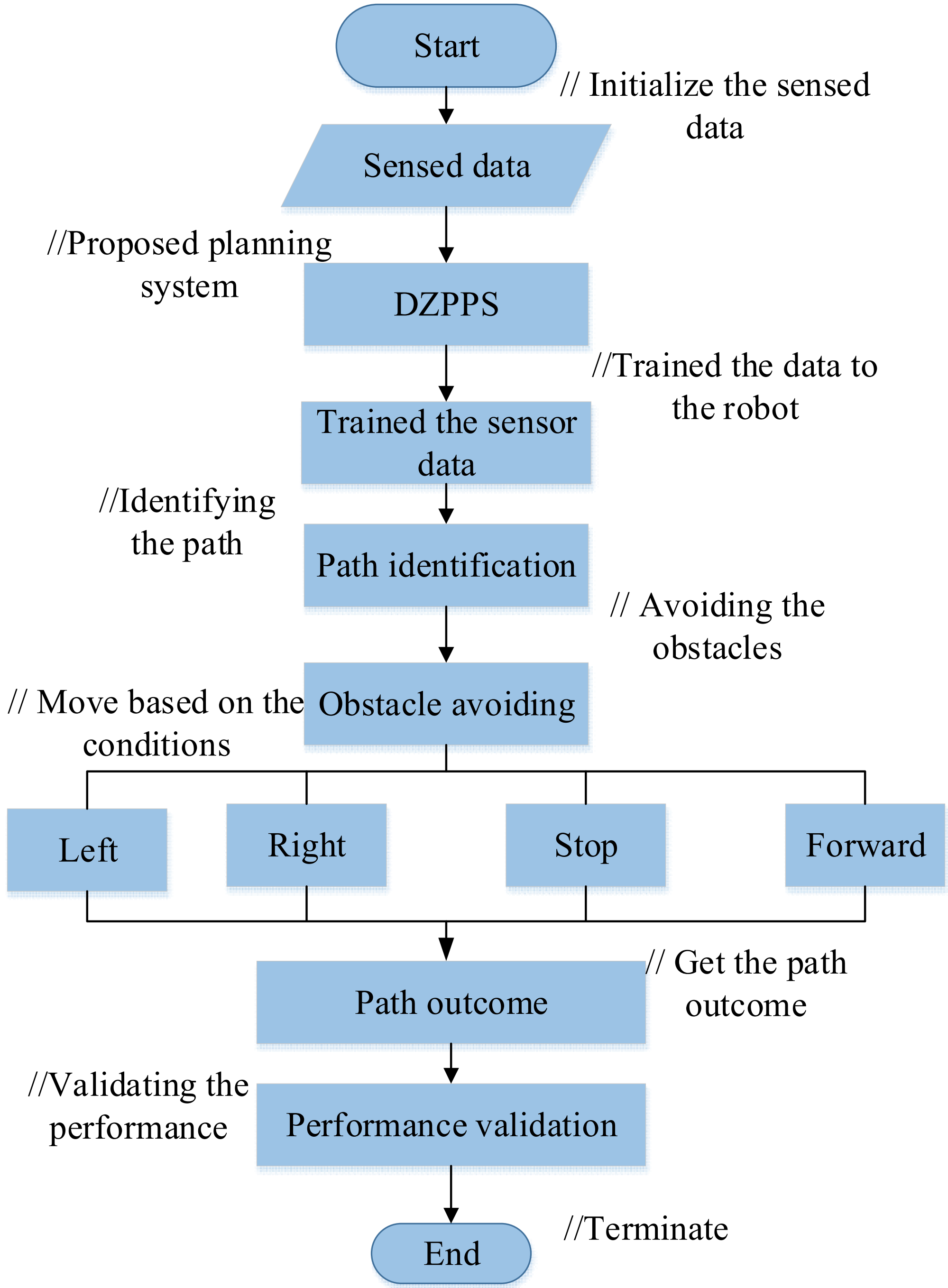

Fig.3: The flow diagram of the proposed DZPPS

Fig.3 shows the phases of the planned procedure. Various metrics have been utilized to assess the degree to which the suggested DZPPS method performed in compliance with the previously mentioned parameters.

4. RESULT AND DISCUSSION

The Python tool running on Windows 10 is used to verify the designed DZPPS. Initially, the sensed data was trained on the robot to identify the path. To get beyond the barriers in the way, use the decision function. The requirements for executing the modelled DZPPS are described in Table 1.

Table 1: Implementation Parameters

Parameters	Description
Programming language	Python
Version	3.7.14
Operating system	Windows 10
Dataset	Sensed data
Network	Zfnet
Algorithm	Decision tree

4.1 Case study

In the context of a warehouse, a robot must find its way from one starting point to another. Obstacles like shelves, cartons, and machines litter the warehouse. The robot must avoid various obstructions to reach the goal as soon as possible. An FPGA is typically utilized to build a real-time search algorithm that determines the shortest route between the origin and the destination while avoiding obstacles. This is the case in most FPGA path planning and obstacle avoidance systems. An additional application for the FPGA is a motion control system that adheres to the predetermined course.

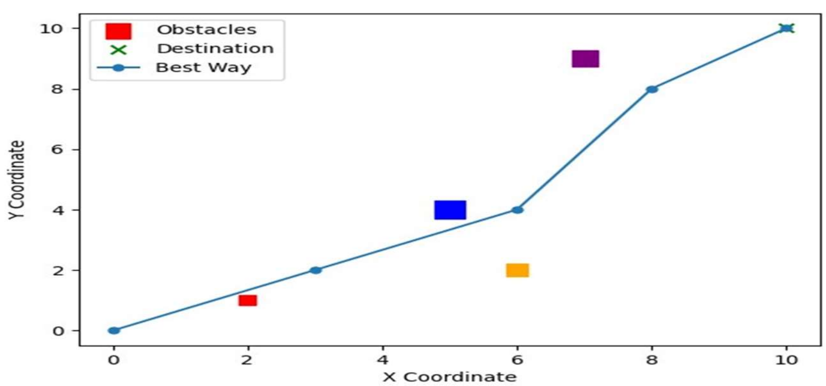

Fig.4: Obstacle Avoiding For Moving Robot From Start To Target Point

Fig. 4 shows the path after avoiding the Obstacle. The diagram is a line graph showing the distance to a destination as a function of the x-coordinate. The y-coordinate represents the distance to the target, while the x-coordinate indicates the distance along the optimal path. And also it shows that there are some obstacles along the way, but the best path avoids them.

4.2 Performance validation

The proposed DZPPS was tested using the sensed dataset and the Python environment. The outcomes were confirmed, and path length, time, and path efficiency were computed. The calculated results are then combined with a few existing models to support the efficiency enhancement. The current methods taken for comparison are Particle Swarm Optimization (PSO) [26], Genetic algorithm (GA) [26], Artificial Bee Colony (ABC) [26], Simultaneous Localization And Mapping (SLAM) [27], Binarized Neural Network (BNN) [28], and Object Recognition and Sorting System (ORSS) [29].

4.2.1 Path length

A robot's overall distance from its beginning location to its goal point is known as its path length in robot path identification. It is a gauge of the robot's path's effectiveness. The robot's journey is more efficient the shorter its length.

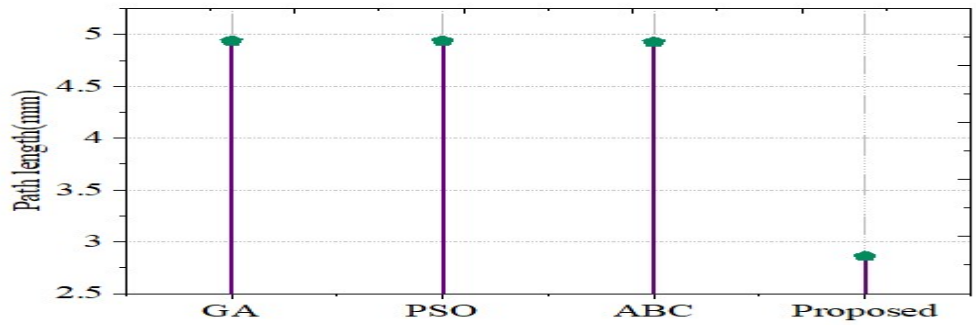

Fig.5: Path Length Comparison

The path length comparison is shown in Fig.5. Here, the proposed DZPPS's path length is verified and compared with the existing methods like PSO, GA and ABC. The path length for the PSO, GA and ABC were 4.94m, 4.94m and 4.93m. The validated proposed model's path length is also 2.86m, which is better than the existing methods. The implemented DZPPS model has demonstrated strong performance in path identification, as seen by the decreased path length.

4.2.2 Execution time

The time required for a robot to find its way from the beginning to its target location while avoiding obstacles is known as the robot route planning and avoidance time. It is a gauge of the robot's path's effectiveness. The robot's way is more efficient in the shorter the time.

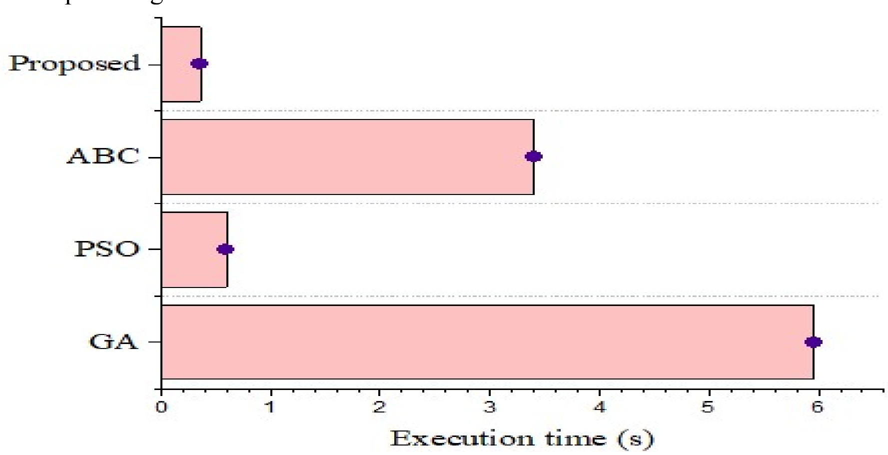

Fig.6: Execution Time Comparison

The execution time assessment is displayed in Fig.6. Here, the PSO method earned a time of 0.60s, GA earned 5.96s, and the ABC model earned 3.40s. The time taken for the proposed DZPPS is 0.36s. The proposed DZPPS method obtained a lower time than the other related models.

4.2.3 Distance from Obstacles

Among the most crucial parameters for robot navigation is distance from obstacles. The robot navigates its surroundings safely and effectively by accurately estimating the distance between blocks. The distance a robot travels to avoid impediments is known as its distance from obstacles.

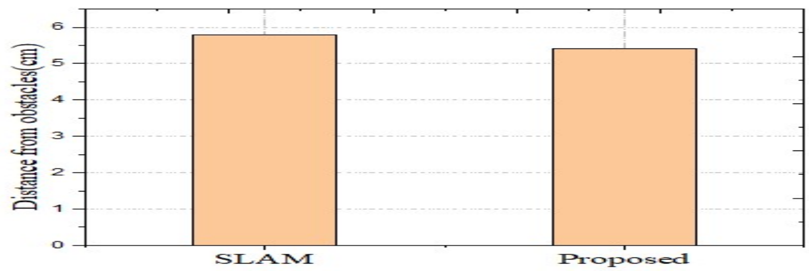

Fig.7: Distance From Obstacles

Fig. 7 displays the comparison of distance from obstacles. It is a crucial navigational parameter for robots since it lets them carefully plan their routes and avoid mishaps. The space for the SLAM method is 5.8cm, and the proposed DZPPS method is 5.41cm. Table 2 shows the overall comparison.

Table 2: Overall Comparison

Methods	Path length (mm)	Execution time (s)	Distance from obstacles (cm)
PSO	4.94	0.60	-
GA	4.94	5.96	
ABC	4.93	3.40	
SLAM	-	-	5.8
Proposed	2.86	0.36	5.41

4.2.4 Flipflop

To store and update intermediate path information in FPGA-based path planning and obstacle avoidance, a flipflop (FF) is essential. A memory element holds the path's current state, allowing the FPGA to make deft decisions based on the environment's cumulative information.

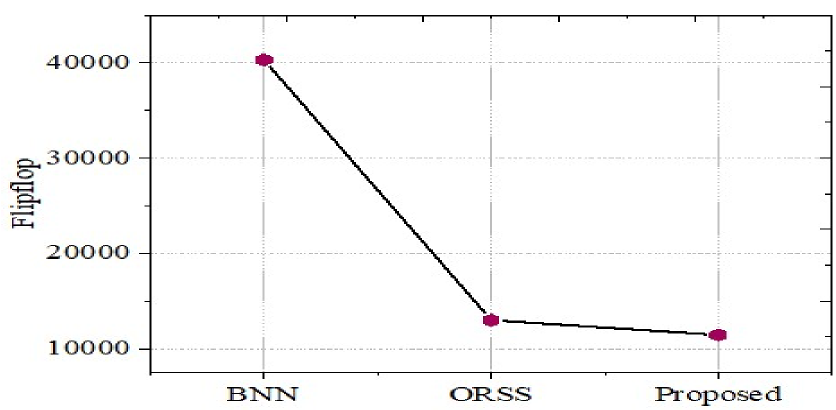

Fig.8: Flipflop Comparison

Fig. 8 shows the flip-flop comparison. Here, the BNN method earned the Flipflop is 40309, and the ORSS method earned 13000 of Flipflop. The Flipflop is taken for the proposed DZPPS is 11452. The proposed DZPPS method obtained a lower Flipflop than the other related models.

4.2.5 Lookup table (LUT)

A lookup table (LUT) is a type of memory used in FPGA-based path planning and obstacle avoidance systems that hold precomputed values for a particular function. Path planning algorithms perform better, and calculations are completed more quickly when LUTs are employed. LUTs can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

be used in path planning to store the distances between several environmental points. Because the FPGA needs to look up the precomputed value rather than doing the computations every time, this can drastically reduce the computational overhead associated with computing distances during path planning. Information can also be stored using LUTs.

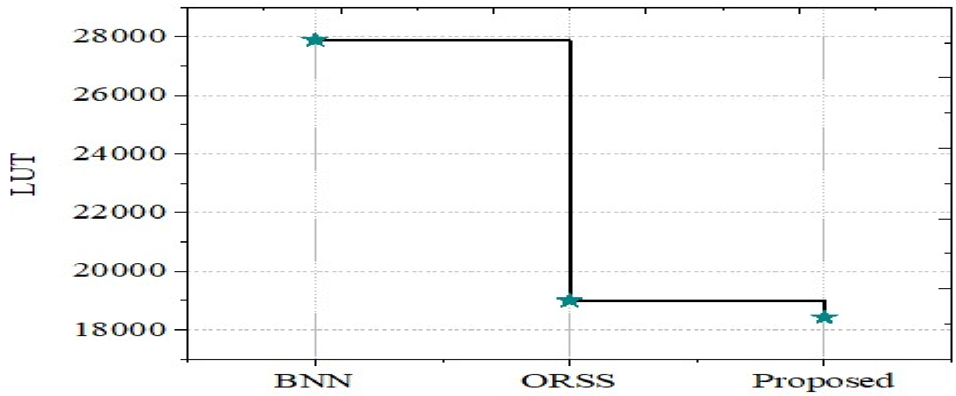

Fig.9: LUT Comparison

The Fig.9 shows the LUT comparison. The BNN method earned the LUT 27887, and the ORSS method made 19000 LUT. The LUT taken for the proposed DZPPS is 18425. The proposed DZPPS method obtained a lower LUT than the other related models. Table 2 describes the overall comparison. Table 3 shows the comprehensive comparison of Flipflop and LUT.

Table 3: Overall Comparison Of FF And LUT

Methods	Flipflop	LUT
BNN	40309	27887
ORSS	13000	19000
Proposed	11452	18425

4.3 Discussion

Difference from prior work:

The proposed DZPPS method attained a lower path length than the existing methods. The model achieved an 11.507m path length. The execution time is 2.0135s. Overall, the DZPPS method performed better compared to the other methods. Table 4 shows the overall effectiveness of the suggested DZPPS.

Table 4: Performance Of The DZPPS

Metrics	Performance
Path length	2.86mm
Execution time	0.36s
Distance from obstacles	5.41cm
Flipflop	11452
LUT	18425

5. CONCLUSION

Autonomous robots' proficiency in route planning and obstacle avoidance is essential for their capacity to function in various dynamic settings. Path planning and obstacle avoidance can be approached in multiple ways, leading to an aimless journey. As a result, our effort produced a unique Decision Zfnet Path Planning System (DZPPS). First, the sensed data was used to teach the robot's behaviour. Next, the robot's course was determined. After that, with the help of the planning system, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

robot was meant to avoid impediments by avoiding them when they were encountered. The decision function is employed to steer clear of obstructions when navigating the robot. The proposed method earned a path length of 11.507m. When compared to the traditional methods, it was improved by 3%. The time taken for execution is 2.0135s; compared to the conventional method, the time was improved by 2%. In future, investigate related architectural and semantic acceleration strategies focusing on various real-time robotics application kernels.

REFERENCES:

[1]. Van Aerschot, Lina, and Jaana Parviainen. "Robots responding to care needs? A multitasking care robot pursued for 25 years, available products offer simple entertainment and instrumental assistance." Ethics and Information Technology 22.3 (2020): 247-256.

[2]. Bardaro, Gianluca, Alessio Antonini, and Enrico Motta. "Robots for elderly care in the home: A landscape analysis and co-design toolkit." International Journal of Social Robotics 14.3 (2022): 657-681.

[3]. Lee, In. "Service robots: A systematic literature review." Electronics 10.21 (2021): 2658.

[4]. Ab Wahab, Mohd Nadhir, et al. "Path planning for mobile robot navigation in unknown indoor environments using hybrid PSOFS algorithm." IEEE Access 8 (2020): 161805-161815.

[5]. Akbari, Younes, et al. "Applications, databases and open computer vision research from drone videos and images: a survey." Artificial Intelligence Review 54 (2021): 3887-3938.

[6]. Teng, Siyu, et al. "Motion planning for autonomous driving: The state of the art and future perspectives." IEEE Transactions on Intelligent Vehicles (2023).

[7]. Fareh, Raouf, et al. "Enhancing path quality of real-time path planning algorithms for mobile robots: A sequential linear paths approach." IEEE Access 8 (2020): 167090-167104.

[8]. Tan, Chee Sheng, Rosmiwati Mohd-Mokhtar, and Mohd Rizal Arshad. "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms." IEEE Access 9 (2021): 119310-119342.

[9]. Xiao, Xuesu, et al. "Motion planning and control for mobile robot navigation using machine learning: a survey." Autonomous Robots 46.5 (2022): 569-597.

[10]. Brito, Bruno, et al. "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments." IEEE Robotics and Automation Letters 6.3 (2021): 4616-4623.

[11]. Wang, Binyu, et al. "Mobile robot path planning in dynamic environments through globally guided reinforcement learning." IEEE Robotics and Automation Letters 5.4 (2020): 6932-6939.

[12]. Karur, Karthik, et al. "A survey of path planning algorithms for mobile robots." Vehicles 3.3 (2021): 448-468.

[13]. Fareh, Raouf, et al. "Enhancing path quality of real-time path planning algorithms for mobile robots: A sequential linear paths approach." IEEE Access 8 (2020): 167090-167104.

[14]. Chen, Yiyang, et al. "A neural network-based navigation approach for autonomous mobile robot systems." Applied Sciences 12.15 (2022): 7796.

[15]. Ravankar, Ankit A., et al. "HPPRM: hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots." IEEE Access 8 (2020): 221743-221766.

[16]. Xiao, Xuesu, et al. "Motion planning and control for mobile robot navigation using machine learning: a survey." Autonomous Robots 46.5 (2022): 569-597.

[17]. Chen, Yanjie, et al. "Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics." Soft Computing 24 (2020): 13979-13995.

[18]. Patel, B. K., et al. "A review: On path planning strategies for navigation of mobile robot." Defence Technology 15.4 (2019): 582-606.

[19]. Schmid, Lukas, et al. "An efficient sampling-based method for online informative path planning in unknown environments." IEEE Robotics and Automation Letters 5.2 (2020): 1500-1507.

[20]. Abdulsaheb, Jaafar Ahmed, and Dheyaa Jasim Kadhim. "Classical and heuristic approaches for mobile robot path planning: A survey." Robotics 12.4 (2023): 93.

[21]. Sharma K, Swarup C, Pandey SK, Kumar A, Doriya R, Singh KU, Singh T. Early Detection of Obstacle to Optimize the Robot Path Planning. Drones. 2022 Sep 20;6(10):265.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22]. Orozco-Rosas U, Picos K, Pantrigo JJ, Montemayor AS, Cuesta-Infante A. Mobile robot path planning using a QAPF learning algorithm for known and unknown environments. IEEE Access. 2022 Aug 8;10:84648-63.
[23]. Shi W, Wang K, Zhao C, Tian M. Obstacle avoidance path planning for the dual-arm robot based on an improved RRT algorithm. Applied Sciences. 2022 Apr 18;12(8):4087.
[24]. Li, Yonggang, et al. "A mobile robot path planning algorithm based on improved a* algorithm and dynamic window approach." IEEE Access 10 (2022): 57736-57747.
[25]. Alarabi, Saleh, Chaomin Luo, and Michael Santora. "A PRM Approach to Path Planning with Obstacle Avoidance of an Autonomous Robot." 2022 8th International Conference on Automation, Robotics and Applications (ICARA). IEEE, 2022.
[26]. Baziyad, Mohammed, et al. "Addressing real-time demands for robotic path planning systems: A routing protocol approach." IEEE Access 9 (2021): 38132-38143.
[27]. Mittal, Rohit, et al. "A novel approach to localized a robot in a given map with optimization using GP-GPU." Recent Trends in Communication and Intelligent Systems: Proceedings of ICRTCIS 2019. Singapore: Springer Singapore, 2020. 157-164.
[28]. Kojima, Akira, and Yohei Nose. "Development of an autonomous driving robot car using FPGA." 2018 International Conference on Field-Programmable Technology (FPT). IEEE, 2018.
[29]. Wang Y, Han Y, Chen J, Wang Z, Zhong Y. An FPGA-Based Hardware Low-Cost, Low-Consumption Target-Recognition and Sorting System. World Electric Vehicle Journal. 2023 Sep 4;14(9):245.
[30]. Zulfiqar, Hasan, et al. "Identification of cyclin protein using gradient boost decision tree algorithm." Computational and structural biotechnology journal 19 (2021): 4123-4131.
[31]. Fu L, Feng Y, Majeed Y, Zhang X, Zhang J, Karkee M, Zhang Q. Kiwifruit detection in field images using Faster R-CNN with ZFNet.
[32]. IFAC-PapersOnLine. 2018 Jan 1;51(17):45-50.