Volume 1, Issue X, October 2022, **pp.** No. 21, pp. 270-282
Submitted 17/10/2022
Final peer reviewed 30/10/2022
Online Publication 7/11/2022
Available Online at http://www.ijortacs.com

# MODELING A DEEP LEARNING AND FUZZY LOGIC-BASED BEHAVIORAL APPROACH FOR AUTONOMOUS NAVIGATION OF A ROBOT IN A GLOBAL POSITIONING SYSTEM (GPS) DENIED ENVIRONMENT

[1]Innocent Ifeanyichukwu Eneh, [2]Princewill Chigozie Ene, [3]Emmanuel C. Obasi
[1,2,3]Enugu State University of Science and Technology (ESUT)

[1]innocent.ifeanyichukwu@esut.edu.ng;
[2]eneh.princewill@esut.edu.ng;[3]oechukwubueze@gmail.com
[1]**DOIs:**https://orcid.org/0000-0001-6116-1607

**Abstract**

***This paper presents the modeling of a deep learning and fuzzy logic behavioral approach for the autonomous navigation of a robot in a Global Positioning System (GPS) denied environment. The study was aimed at addressing the optimization problems experienced by mobile robots due to the dynamics of an environment as a result of global positioning system unavailability. This problem was addressed by collecting data from the workspace environment and then training a deep neural network model to generate a cognitive algorithm that was used for intelligent Simultaneous Localization and Mapping (SLAM) using the fuzzy logic approach. The algorithm was integrated into a deferential wheel drive robot using Simulink and was tested. The result showed an accuracy of 99.80% and a loss function of 0.20%, which implied good training performance and SLAM intelligence. The deep fuzzy algorithm when integrated into the robot and tested in a dynamic environment that has no GPS was able to intelligently maneuver obstacles in the workspace.***

***Keywords: Deep learning, GPS, Mobile Robot, Fuzzy, SLAM, dynamic environment.***

## 1. INTRODUCTION

Mobile robots are becoming increasingly present in everyday life and this can be accredited to the fast development in microelectronics, communication, automation, navigation, and robotics they have rapidly changed the working environment in a way that mobile robots are a part of modern life (Oltean, 2019). The design of ground robots is a big research field and it presents very large industrial stakes as these robots are used increasingly in the industry as means of transport or for inspection (Melik and Slimane, 2016).

Studies show that navigation is a critical topic in the design and development of autonomous robots because a robot can run into impediments on a transitional path. A system is intelligent when it can assess a situation and use the knowledge it gathers to take wise judgments (in this case, avoiding obstacles and maintaining a smooth path toward a target). That is, for a robot to be able to sense its environment, process data gathered from it, identify an obstruction in front of it, interpret what is seen, and then make suitable decisions, it has to be intelligent. The technique by which a robot moves independently through a collection of coordinates of locations without running into other things or becoming lost is known as navigation. One way to implement this is through artificial intelligence(Klancar et al., 2017).Artificial intelligence is the capacity of a robot to develop the ability to process information from its surroundings and make decisions from that through mimicking human intelligence. Algorithms that detect and avoid obstacles along a path would typically not be required if a robot's line of travel were designed geometrically, from its current location to its goal, and if there were strong assumptions of certainty that there would be no impediments along its path. The development of algorithms to help a robot recognize and avoid obstacles started in response to the reality that an autonomous robot may be in a situation where it has no prior knowledge of the challenges it may face (Wang et al., 2015).

Robots can avoid objects that are not their intended targets by using strategies known as obstacle avoidance methodologies, which allow them to keep moving in the direction they were headed. It entails directing the robot's path to get around unforeseen obstructions. The robot is seen as an autonomous item and might be visualized moving through a potential field produced by the objective and by the environmental impediments. In other words, it might introduce the idea of a dipole, where the objective produces an attracted potential while other objects provide a repulsive potential. The past behaviour of robots close to barriers can be used to anticipate how a robot will act in the future when it approaches an impediment (and as a result, it is possible to compute the repulsive potential offline). A motion control strategy that establishes the robot's velocity vector to move it toward the goal while avoiding obstacles is also included in the potential field planning approach (Kozlowski and Pazderski, 2004).

In this paper, interest in the autonomous navigation problem of a mobile robot is subjected to perform line following in a partially-known environment that is not GPS enabled. It is partially known because the environment may include unexpected static or dynamic obstacles. The environment is called a safe zone when no object is found in its path and a disaster zone when an obstacle is found in its path (Kozlowski and Pazderski, 2004). The safe zone constitutes a line drawn on the floor in a factory from origin to destination. When an obstacle is found on the line it becomes a disaster zone and the robot stops in front of it and sends an audible message rather than colliding with the object. Ordinarily, in the industry, on such lines, pedestrians or other objects are not to be found there.

### 1.2 Problem Statement

In most autonomous system, cognitive intelligence of their environment is achieved when trained with the GPS coordinates of the environment, but in cases where these GPS data are not available, the robot gets stranded and are often affected by obstacles within the environment. The technical problem has resulted in other social and economic problems like the poor efficiency of mobile robots, limited application of the robot, etc; hence, this paper proposesa solution to this problem using deep learning and fuzzy logic techniques.

## 2. LITERATURE REVIEW

Eneh et al. (2019) presented the application ofdeep learning for autonomous navigation and SLAM operation of a holonomic robot. The study used a convolutional neural network to model a control system that allows the robot navigates smoothly within a dynamic environment. Despite the success, there is a need for cognitive learning in the robot.

Hank and Haddad (2016) proposed a hybrid strategy that is specially designed to handle the autonomous-navigation issue of a mobile robot that is required to complete an emergency task in a poorly known area. Under the limits of the robot's capabilities and known and unknown obstacles, this navigation challenge necessitated a solution that can produce a quick execution time and is flexible enough to handle errors in the known sections of the environment (unexpected obstacles). The work incorporated an offline task-neutral pre-processing stage that is only used once for a certain robot in a specific setting. Its goal was to create a roadmap of close-to-time-optimal reference routes inside the specified zones. The task was carried out online using a combination of reactive navigation and trajectory tracking, with seamless changes between the two modes of navigation.

Cheng et al., (2018) presented the Autonomous Navigation of Mobile Robots in Human Environments. Their study showed that service robots are -used in more and more indoor environments. They recognize the movements of humans in such an environment as dynamic obstacles and proposed ways that robots can overcome such obstacles.

Melik and Slimane (2016) proposed a Fuzzy logic controller for the autonomous navigation of a mobile robot-type tricycle in a partially known environment. Two controllers are developed, the first for free navigation and the second for navigation with avoidance of obstacles present in the environment of a mobile robot using an obstacles detection module. The selected fuzzy controller for the various missions of the mobile robot was Takagi-SugenoType of Order Zero (TS0).

Oltean (2019) presented a mobile robot platform with a fixed four-wheel arrangement chassis and an electronic system built around the Raspberry Pi and Arduino Uno Interfaces. The mobile platform satisfies some fundamental design criteria for initial development by being a low-cost option that is incredibly dependable and expandable.It is proposed for teaching (as a didactic stand for microcontrollers, electronics, automation, and robotics), as well as for research projects to examine various mapping, localization, navigation, obstacle detection, and transport algorithms. The mobile robot platform could serve as a jumping-off point for further research, and some potential uses included autonomous guided robots for indoor environments, medical applications to help patients, military uses, packaging and organizing pallets in warehouses, and transportation of waste materials, laundry, food, and pharmaceuticals or other materials.

## 3. METHODOLOGY

The methodology used an artificial neural network and data collected from a dynamic environment based on three dimensions (3D) sensor perception to develop a model of thecognitive SLAM process and then integrate it into a mobile robot model for obstacle maneuvering in a dynamic environment. The data collected was utilized to train a neural network control system and then embedded on the mobile robot for SLA

operationjumping-off using Robot operating system, Simulink, control system toolbox, and neural network toolbox.

## 4. MODEL OF A DIFFERENTIAL DRIVE ROBOT

The differential drive robot is a two-wheeled drive robot with independent actuators for each wheel. The motion vector of the robot is the sum of the independent wheel motions. In a differential drive robot, the motion of each wheel is controlled by one DC motor. The DC motor receives voltages as input and gives out torque as output. The torque produced on the axis of the robot causes the wheels of the robot to spin. The movement of the wheel will produce the pose of the robot. The pose of the robot is the orientation of the body frame concerning the world frame as seen in figure 1. To determine how the robot will move when it receives a sequence of commands (which is called forward kinematics) and what command is required for the robot to

DC motor hasthe desired movement (inverse kinematics) the model of figure 1 is developed on the following assumptions (Kozlowski and Pazderski, 2004):

ii. The robot is symmetrical along its longitudinal axis ($x^r$). That is, it has equal distance wheels (axial length = 2L), the wheels are identical ($R_l$ = $R_r$) and the centre of mass of the robot is at distance c from A as seen in Figure 1.

iii. The robot is a rigid body. That is, the distance between any two points of the robot does not change.

i. The robot has two frames: The body frame ($x^r$, $y^r$) and the world Frame ($x^w$, $y^w$), and the body frame moves with respect to the world frame.

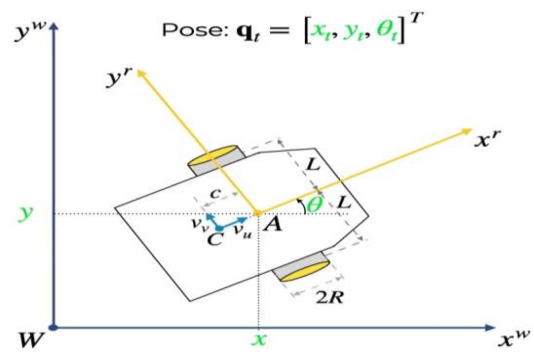


Figure 1: Differential drive robot (Kozlowski and Pazderski, 2004).

Rigid body rotation: $\omega=[0,0,\theta']^T$(.1)

The velocity of point A in the body frame is given as $V_A^r = \begin{bmatrix} V_u \\ V_v - C\theta' \end{bmatrix}$ (.2)

and$(V_v - C\theta')$ is the lateral velocity at point A in the body frame. C is the centre of the mass of gravity. From the body frame, the velocity of the world frame can be obtained using a rotational matrix(Kozlowski and Pazderski, 2004).

$$V_A^w = \begin{bmatrix} x' \\ y' \end{bmatrix} = R_\theta V_A^r = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} V_u \\ V_v - C\theta' \end{bmatrix} \quad (3)$$

The rotational matrix is defined by the orientation angle $\theta$.

General kinematics equations (Kozlowski and Pazderski, 2004):

$$x_t' = V_{ut}cos\theta - (V_{vt} - C\theta')sin\theta \quad (4)$$

$$y_t' = V_{ut}sin\theta - (V_{vt} - C\theta')cos\theta \quad (5)$$

$$\theta_t' = \omega_t \quad (6)$$

Further, it is assumed that the robot has no lateral movement, which means that the robot does not skid. This makes the lateral velocity in the body frame at point A to be zero.

$$\text{Hence, } V_A^r = V_u \quad (7)$$

And also, that each wheel travels a distance equal to its circumference for every full rotation means no slipping occurs during the movement of the wheels(Kozlowski and Pazderski, 2004):

$$\Delta x = 2\pi R \quad (8)$$

With these assumptions, at every instance the linear velocity of the wheel is given by the product of the radius of the wheel and the angular velocity(Kozlowski and Pazderski, 2004):

$$V_{ur} = \omega_{ur}R \quad (9)$$

And equations (10) and (11) become:

$$x_t' = V_{ut}cos\theta \quad (10)$$

$$y_t' = V_{ut}sin\theta \quad (11)$$

With the no-sliding assumption, the relationship between the robot and the world frame is simplified. The no-slipping assumption allows the relationship between the velocity of the wheel and that of the robot to be determined.

Figure 2 shows the instantaneous centre of curvature (ICC). It is the only point on a rotation field that does not move.

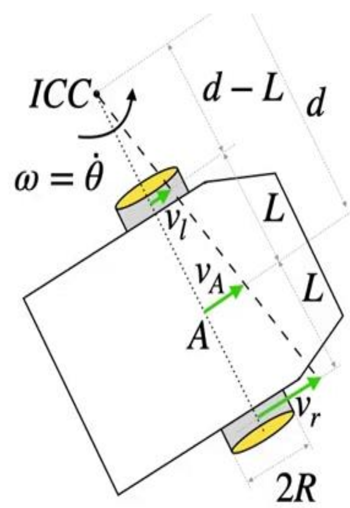


Figure 2 Instantaneous centre of curvature.

$$d = L\,\frac{V_r + V_l}{V_r - V_l} \tag{12}$$

$$\begin{bmatrix} V_l = \omega(d - L) \\ V_A = \omega d \\ V_r = \omega(d + L) \end{bmatrix} \tag{13}$$

If $V_l = V_r$ no turn. The robot moves in a straight line. If $V_l = -V_r$ the robot will rotate on the spot. If $V_l = 0, (V_r = 0)$ the robot will turn on the wheel. The centre of curvature, in this case, lies on the position of the static wheel.

**Forward Kinematics:**

$$V_A = \frac{V_r + V_l}{2} \tag{14}$$

$$\omega = \frac{R}{2L}\,(\emptyset'_r - \emptyset'_l) \qquad (\text{`}15)$$

$$\begin{bmatrix} V_A \\ \omega \end{bmatrix} = \frac{R}{2}\begin{bmatrix} 1 & 1 \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix}\begin{bmatrix} \emptyset'_r \\ \emptyset'_l \end{bmatrix}$$

$\emptyset$ is the angular velocity of the wheel, $\omega$ is the angular velocity of the robot. $V_A$ is the linear velocity of the robot and $V_l$ *and* $V_r$ are linear velocities of wheels.

Pose-to-wheel commands can now be mapped to obtain the forward kinematics of the differential drive robot.

$$q'_t = \frac{R}{2}\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix}\begin{bmatrix} \emptyset'_r \\ \emptyset'_l \end{bmatrix} \tag{16}$$

**Inverse Kinematics**

$$\begin{bmatrix} \emptyset'_r \\ \emptyset'_l \end{bmatrix} = \frac{1}{R}\begin{bmatrix} 1 & L \\ 1 & -L \end{bmatrix}\begin{bmatrix} V_A \\ \omega \end{bmatrix} \tag{17}$$

### 4.1 Formulation of the SLAM problem

Localization is the ability of a robot to know where it is on a map, and one way to do this is by using encoder sensors. Encoder sensors are used to compute the number of wheel rotations of the robot to determine the distance traveled. In this project, a Simulink model is used to carry out the localization of the robot. One of the ways to determine the position of a robot, in a GPS-denied environment, is to use the distance traveled by the robot. This method is called dead reckoning and it falls under the problem of knowing where the robot is located.

An encoder is a device connected to the vehicle's wheel to determine the number of wheel rotations. It has ticks, and the complete rotation of the ticks is $360^0$. The encoder is used to build an odometer-a device that is used to calculate distance traveled.

$$\text{Number of wheel rotation encoder} = \frac{\text{total number of encoder ticks}}{\text{total count per encoder}} \tag{18}$$

Distance traveled = Number of wheel rotations x circumference of wheel (19)

$$\text{Distance} = \frac{\text{total number of encoder ticks}}{\text{total count per encoder}} \text{ X circumference of wheel} \tag{20}$$

In equation (20), the input is the total number of encoder ticks, the output is distance and circumference and the total count per encoder are constants. With this, a Simulink model is developed with a simple algorithm. The parameter used is given in Table 1.

**Table 1 Parameters used in distance travel model.**

| Parameters | Values |
|---|---|
| Distance travel | 3 m |
| Robot wheel radius | 0.052 m |
| Ticks per rotation | 627.6 |
| Axle length | 0.27 m |

### 4.2. Data collection

The sensor used in detecting obstacles in this paper is the Lidar and proximity. The lidar sensor was used for the environmental scanning in the size of 120 by 120, while the proximity sensor was used for the mapping of the position of objects along the line of sight.

### 4. Pathfinding problems

The designed path for this robot is a line. This is done on MATLAB with an app called simulation map generator. On this app, an image with a line on it is uploaded and then processed into a grey image.After the upload of the image is done, Image prepossessing is carried out with MATLAB. The pre-processing carried out is image sectioning in which the line on the image is made black and the environment is made white. In this way, the image now has only the line in black and the environment in white. Black areas seen on any other part of the environment are taken as noise, but as they are not so close to the path of the robot, they do not pose much of a problem to the robot.

### 5. Basic Model of Neural Network

To develop the neural network control system, the data collected from the sensor was used to train neural network architecture utilized by Schocken and Ariav (2019) as shown in figure 3 with the various components which are the weight $w_n$, bias ($v_k$), activation function $\theta_k$is the tangent hyperbolic and sigmoid function and output $y_k$;

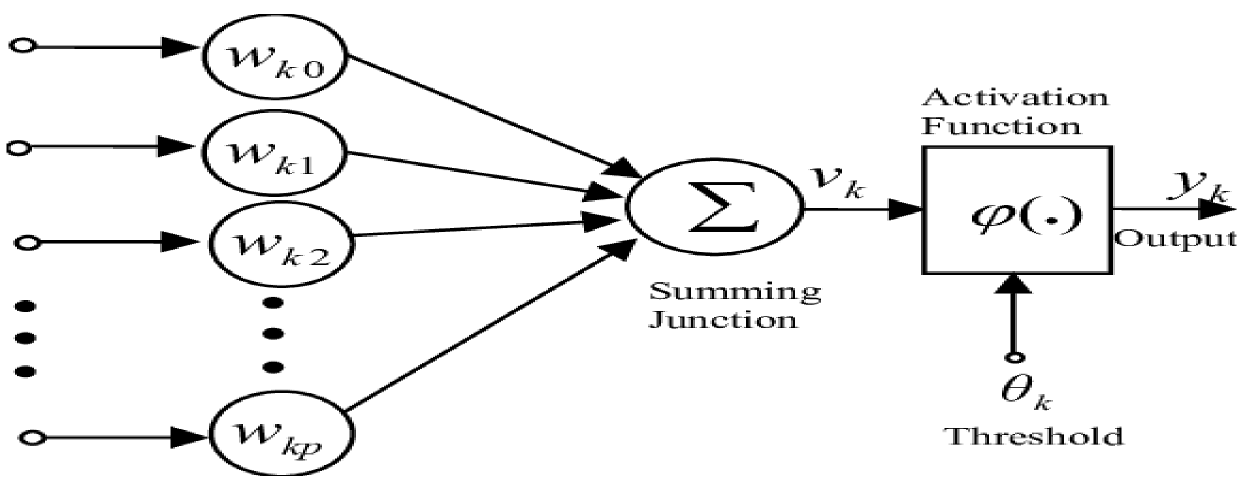


Figure 3: Architectural model of a basic neural network.

The deep learning model used for the development of the intelligent algorithm is the Convolutional Neural Network (CNN) adopted by Eneh et al. (2019). The CNN was reconfigured with three convolutional layers and a single fully connected layer. The workflow chart of the CNN was presented in figure 4;

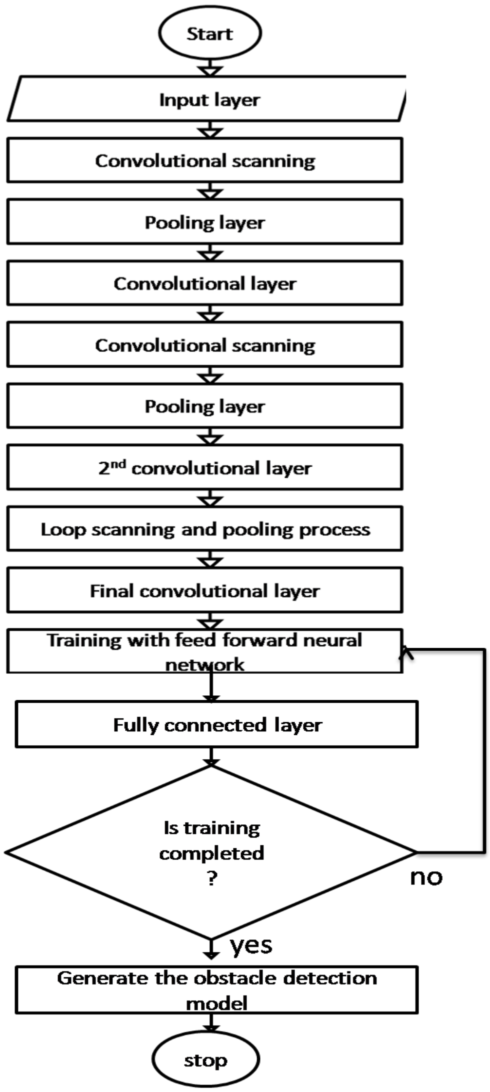


Figure 4: CNN Flowchart

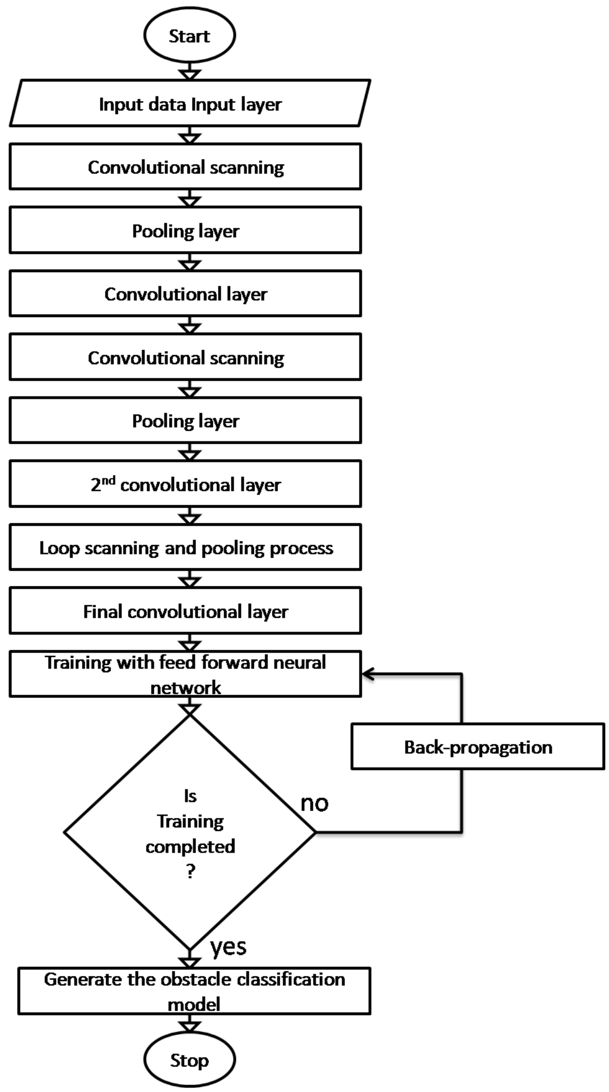


Figure 5: Training flowchart of the CNN

The CNN presented in figure 4 was made of four major layers which are the input layer, the convolutional layer, the fully connected layer, andthen the output layer. The input layer was used to dimension the image into the 120 by 120 and color channel of 3. The filter size used is 5 by 3 to perform convolutional scanning on the image and then the average pooling technique was used to extract the features maps from the strides to form the convolutional layer. The process continues until the final convolutional layer, then the feature maps are flattened and fed to the fully connected layer for training with a feed-forward neural network to generate the obstacle classification algorithm.Figure 5 presented the training of the CNN, showing how data was loaded into the CNN and then trained to generate the obstacle classification model.

### The fuzzy logic

Having developed the CNN model which was used by the root for SLAM, the control to prevent collision with obstacle was achieved with fuzzy logic. Figure 6 presented a basic fuzzy logic model with the fuzzifier at the input, inference engine, and rule-baseddefuzzifier. The fuzzy logic mode in figure 6 was reconfigured using the deep learning-based classification model developed as shown in figure 7;

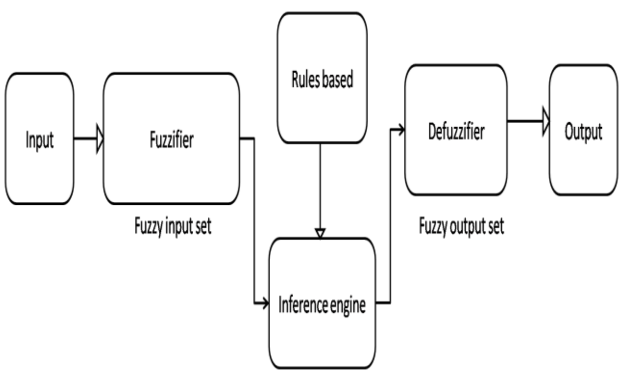


Figure 6: Basic fuzzy logic model

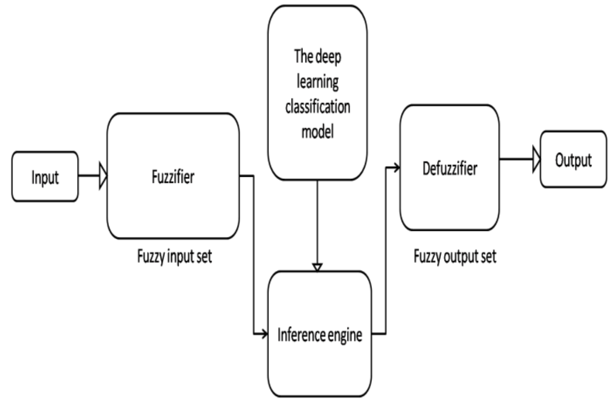


Figure 7: Deep fuzzy model of the robotic control

In the figure deep fuzzy model developed, the rulebase in the conventional fuzzy approach was improved using a deep learning-based classification model. Here the input from the sensor was fuzzified and trained at the inference engine, using the classification model to detect obstacles and then maneuver. The algorithm of the deep fuzzy model for the robotic control was presented as;

### Deep fuzzy algorithm

1. ***Start***
2. ***Data collection from Lidar sensor (a)***
3. ***Set initial robot coordinates as (v) and (u)***
4. ***Data collection from proximity sensor % Set proximity coordinated as x and y axis***
5. ***Fuzzy set***
6. ***Deep learning classification model***
7. ***Training with inference engine***
8. ***If***
9. ***a is classified as = true***
10. ***Get new coordinates as (x; y) – (v ; u) as n***
11. ***Continue navigation at n+1***
12. ***Else***

13. *Continue navigation at x and y*
14. *Return*
15. *Stop*

## 6. SYSTEM IMPLEMENTATION

The models and algorithms developed were implemented using a robot operating system, control system toolbox, image acquisition, optimization toolbox, deep learning toolbox, fuzzy toolbox, and Simulink. The Simulink model of the robot was presented in figure 8;

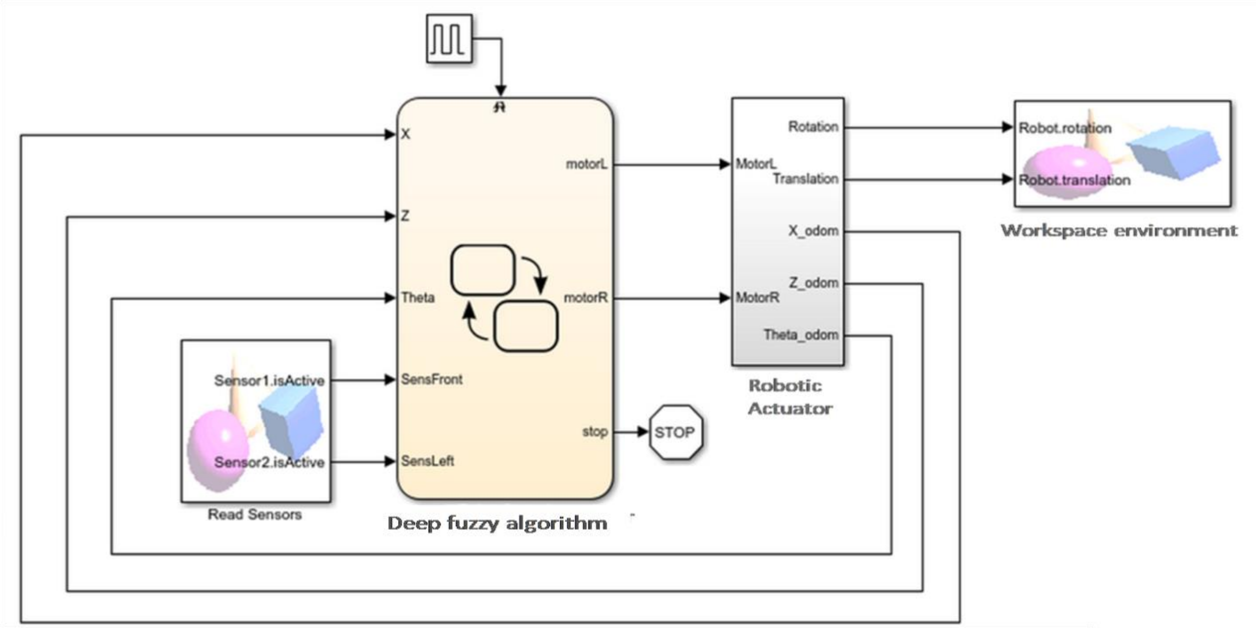


Figure 8: Simulink model of the robot

The model showed how the sensor which collects data from the environment needs a deep fuzzy algorithm for intelligent SLAM operation and then free navigation. The next section presented the simulation performance of the mobile robot developed with the intelligent control system.

## 7. RESULT AND DISCUSSIONS

The data used on the ANN are divided into three sets: training, validation, and testing. After every iteration step, the network is validated with the validation data and finally, when better performance is obtained by the network, it is then tested with the test data. In other words, the ANN is turned with the validation data, and when better performance is achieved, it is then tested with the test data. The purpose of having a set of validation and test data is to have an unbiased network. Figure 9represents the result obtained after training the deep learning algorithm considering the accuracy and loss function.

Figure 10 also presented the result of the system integration which showed how the robot operated with the deep fuzzy algorithm to maneuver obstacles in a dynamic environment without GPS.

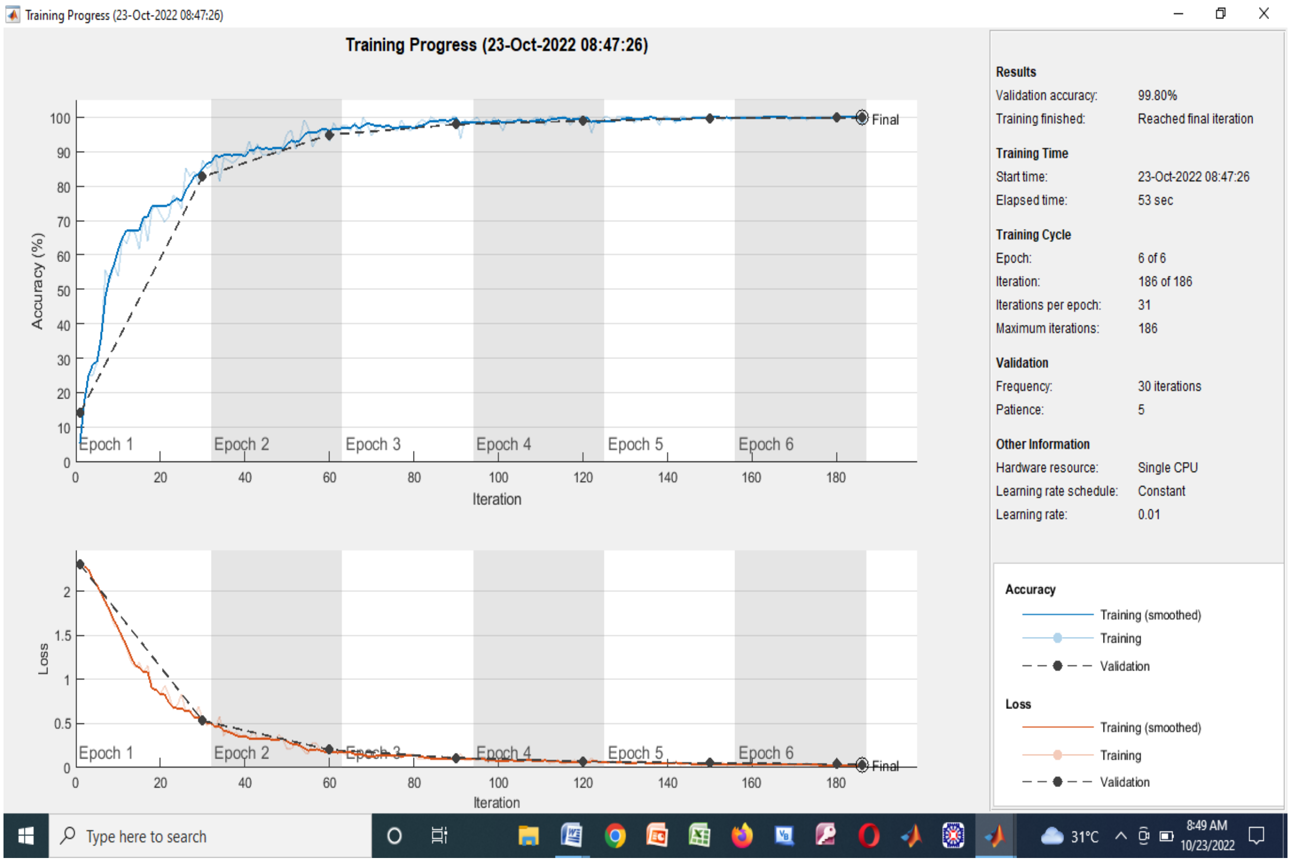


Figure 9: The training result of the deep learning algorithm


Figure 10: Integrated robot navigation with deep fuzzy controller

Figure 9 presented the training result of the deep learning algorithm, the result showed that the accuracy achieved is 99.80% and the loss function is 0.20%. The implication of the result showed that the robot correctly has knowledge of the objects within the propagation path and make avoid obstacle without GPS information of the environment.Figure10 is the robot wheel behavior with the deep fuzzy controller, showing how the robot navigates while collecting data from the environment and then train to make cognitive SLAM and avoid obstacles. The two colors in figure 11 represent the behaviorof the two motor wheels controlled by the robot controllers in the two different cases. Showing how the robot turns its wheel with respect to signals received from the controllers based on the informed intelligence from the SLAM process.

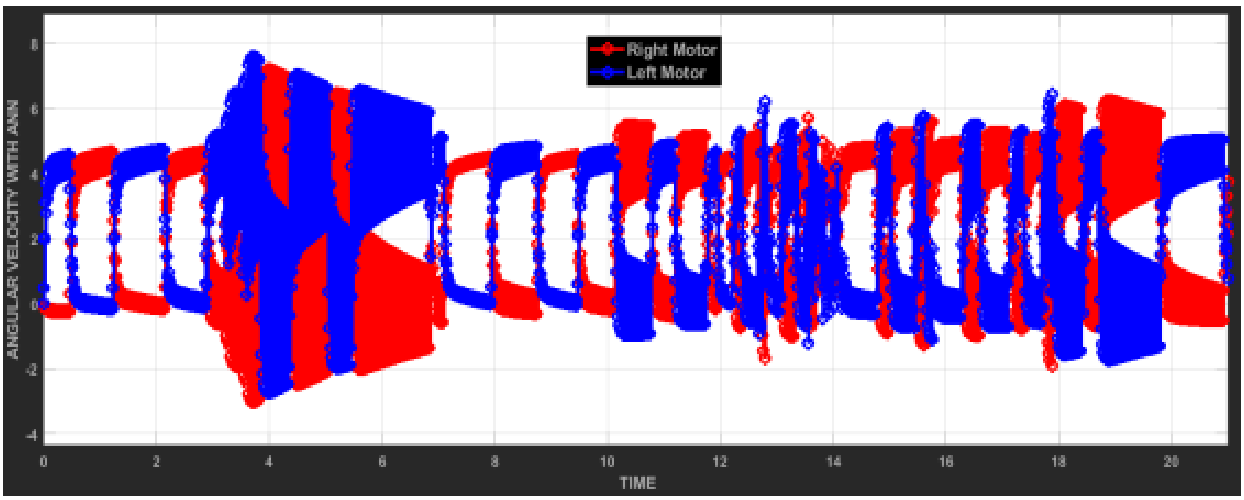


Figure 11: Behavior of Robot wheel with ANN controller

## 8. CONCLUSION

In this paper, the autonomous navigation of a robotis modeled with a differential drive robot in a GPS-denied environment. It involves navigation functions such as perception of the environment for obstacles with an ultrasonic sensor, and path tracking using the line following technique for localization. The robot SLAM operation was optimized using cognitive intelligence via training of a neural network control system. The result when tested showed that the root was able to correctly learn its environment via cognitive SLAM and then freely navigate to carry out a task.

## 9. CONTRIBUTION

This research has developed a differential drive robot with cognitive SLAM intelligence using adeep fuzzy control system, for the intelligent maneuvering of obstacles within a dynamic workspace when GPS is denied.

## 10. REFERENCES


Cheng, J., Cheng, H., Meng, M. Q. H., and Zhang, H. (2018). Autonomous Navigation by Mobile Robots in Human Environments: A Survey. 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018, 1981–1986. https://doi.org/10.1109/ROBIO.2018.8665075

Eneh Princewill C., Eneh Innocent I., EgoigweSochima V., and Ebere Chidi U. (2019), "Deep Artificial Neural Network Based Obstacle Detection and Avoidance for a Non-Holonomic Mobile Robot" International Research Journal of Applied Sciences, Engineering, and Technology, Vol.5, No.1; January-2019; ISSN (1573-1405); p –ISSN 0920-5691. Available on www.cird.online/IRJASET.

Hank, M., and Haddad, M. (2016). A hybrid approach for autonomous navigation of mobile robots in partially-known environments. Robotics and Autonomous Systems, 86, 113–127. https://doi.org/10.1016/j.robot.2016.09.009

Klančar, G., Zdešar, A., Blažič, S., and Škrjanc, I. (2017). *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Butterworth-Heinemann. ISBN 9780128042045. https://doi.org/10.1016/B978-0-12-804204-5.00001-9.

Kozlowski, K., and Pazderski, D. (2004). Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, *14*(4), 477–496. https://doi.org/10.1016/j.compag.2011.12.009

Melik, N., and Slimane, N. (2016). Autonomous navigation with obstacle avoidance of tricycle mobile robot based on fuzzy controller. 2015 4th International Conference on Electrical Engineering, ICEE 2015, 1. https://doi.org/10.1109/INTEE.2015.7416799

Oltean, S. E. (2019). Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation. Procedia Manufacturing, 32, 572–577. https://doi.org/10.1016/j.promfg.2019.02.254

Princewill Chigozie Ene (2022)"The Kinematic Modeling Of A Four Mecanum Wheel Robot For Environment Mapping And Navigation"[IJORTACS] Volume 1, Issue VIII, August 2022, No. 15, pp. 194-209

Schocken S, Ariav G (2019). Neural Networks for Decision Support Systems: Problems and Opportunities. New York: Center for Research on Information Systems, Stern School of Business, New York University.

Wang, T., Wu, Y., Liang, J., Han, C., Chen, J., and Zhao, Q. (2015). Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor. *Sensors (Switzerland)*, *15*(5), 9681–9702. https://doi.org/10.3390/s150509681

Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017). "ImageNet classification with deep convolutional neural networks". Communications of the ACM. **60** (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782; 195908774.