

Improving the Control of Autonomous Navigation of a Robot with Artificial Neural Network for Optimum Performance

Obasi Emmanuel Chukwubueze, Eneh Innocent Ifeanyichukwu, Ene Princewill Chigozie *

Electrical and Electronic Engineering Department, Enugu State University of Science and Technology (ESUT), Enugu, Nigeria

Email address:

engrprofobasi@gmail.com (Obasi Emmanuel Chukwubueze), innocent.ifeanyichukwu@esut.edu.ng (Eneh Innocent Ifeanyichukwu), eneh.princewill@esut.edu.ng (Ene Princewill Chigozie)

*Corresponding author

To cite this article:

Obasi Emmanuel Chukwubueze, Eneh Innocent Ifeanyichukwu, Ene Princewill Chigozie. (2024). Improving the Control of Autonomous Navigation of a Robot with Artificial Neural Network for Optimum Performance. *Engineering Science*, 9(1), 12-20.

<https://doi.org/10.11648/j.es.20240901.13>

Received: July 4, 2023; **Accepted:** July 25, 2023; **Published:** January 8, 2024

Abstract: In the field of autonomous robotics, enhancing the navigation system of robots is a crucial aspect that directly impacts their performance. This study presents a novel approach to addressing this challenge with an Artificial Neural Network (ANN) model. The research focuses on improving the navigation capabilities of a differential drive robot using the line-following method for route tracking and the dead reckoning technique for localization. It investigates a differential drive robot model controlled with a PID controller and derives the transfer function of the PID model. Through simulations, it becomes apparent that the PID model exhibits a continuous overshoot in its response, which negatively affects the behaviour of the robot's wheels. Ordinarily, continuous manual tuning will be required to correctly tune the PID controller to a value where the overshoots will be negligible, and this could be onerous. To overcome this limitation, an ANN controller is proposed, leveraging the learning capabilities of the neural network. Data from the PID controller transfer function is utilized to train the ANN model, enabling it to understand patterns and relationships. The ANN controller is then substituted in place of the PID controller in the simulation. The results showcase a remarkable 13.1% improvement in the robot's wheel response, highlighting the transformative potential of this approach for revolutionizing autonomous robot navigation in industrial applications. By using the transfer function of the PID model to train an ANN model, this study offers a powerful framework for enhancing the navigation performance of a differential drive autonomous robot and shows performance improvements in control, flexibility, and adaptation to changing conditions. These discoveries have significant ramifications for the industry and will pave the way for intelligent and effective autonomous robot navigation systems. The research provides a comprehensive understanding of the challenges associated with the differential drive robot model controlled with a PID controller and offers a robust approach to how this can be alleviated. The significance of this study lies in its ability to address the continuous overshoot issue observed in the PID controller's response by training an ANN controller with data from the PID controller. The proposed approach minimizes overshoot and improves the robot's wheel response, ultimately enhancing its navigation capabilities. Overall, this study demonstrates the potential of an ANN model to revolutionize autonomous robot navigation in industrial applications. The notable improvement achieved in the robot's wheel response validates the effectiveness of this approach. Future research can further optimize this integrated approach in real-world scenarios, leading to intelligent and efficient autonomous robot navigation systems across diverse industrial settings.

Keywords: Transfer Function, PID Controller, Artificial Neural Network (ANN), Route Tracking, Localization, Over-Shoots, Performance Improvements

1. Introduction

Because of the rapid advancements in microelectronics,

communication, automation, navigation, and robotics, mobile robots are becoming more prevalent in daily life [13]. These advancements have also drastically altered the working

environment. Robots can be remotely operated, somewhat autonomous, or fully autonomous, depending on the intended usage [6]. As these robots are increasingly utilized in the industry as a means of transportation or for inspection, the design of ground robotics is a huge research area with very high industrial stakes [12]. Studies reveal that because a robot may encounter obstacles on a transitional path, navigation is a crucial component in the design and development of autonomous robotics. It is essential to consider the coordination, interaction, and coupling of a robot's various units to effectively control its functionalities, mobility, and manipulation. This involves taking into account the kinematics (movement-related properties), dynamics (interactions with external forces), and connectivity between various components [14]. When a system can analyze a situation and make informed decisions (in this example, avoiding obstacles and maintaining a smooth path toward a destination), it is said to be intelligent. That is, a robot must be intelligent enough to detect its surroundings, evaluate the data collected from it, recognize an obstacle in front of it, interpret what is observed, and then make appropriate judgments. Navigation is the process through which a robot moves autonomously across a grid of coordinates without colliding with anything or becoming lost. Artificial intelligence is one method of putting this into practice.

2. Related Works

Finding a route that moves the robot from a starting point in the environment to a target while avoiding collisions with obstacles is the responsibility of the autonomous controller [4]. The navigation of mobile robots can be approached in a variety of ways. The most popular navigational algorithms for all types of mobile robots are go-to-goal, avoiding obstacles, wall-following, and path planning [11]. Wheel encoders are employed for robot localization, while sonar range sensors are used as the sensing elements for obstacle detection. The issue of autonomous navigation in agricultural fields was examined [5] and it led to the suggestion that to create a hybrid topological map, it will need a localization and mapping framework based on semantic place classification and key location estimates. In indoor scenes, a reliable recognition model is essential for improved indoor navigation performance, enabling disabled individuals to interact effectively with their environment, while indoor scene understanding and object classification are crucial for various vision applications like robot navigation assistance and indoor object recognition [1]. The Vector Field Histogram is one of the techniques for navigation, it creates a polar histogram of the space occupancy near the robot so that the robot's steering is then adjusted to point in the direction of the sector of the polar histogram with the lowest density of polar obstacles after the polar histogram built around its current location has been validated [3]. Developing an internal representation of space that is supported by recognized landmarks and powerful visual processing that concurrently supports continual self-localization ("I am here") and a representation of the goal ("I am heading there") is a difficult undertaking that is required for navigation [9]. This,

however, is possible, with artificial intelligence (AI) and related technologies since autonomous robots can automate a wide range of labour-intensive jobs in the factory environment and raise output. The number of robots deployed in the manufacturing business has expanded quickly, and this trend is anticipated to continue in the future. Swarm optimization and sensor deployment can be applied for autonomous navigation and obstacle avoidance of an omnidirectional mobile robot [2]. This involved environmental modelling and path search algorithms based on optimization criteria to achieve the best path. Environmental data gathered by a Forward-looking Sonar (FLS), can be utilized in predictive guidance obstacle avoidance algorithm (PGOA) in unknown environments for autonomous underwater vehicles (AUVs) that must adapt to multiple complex obstacle environments [3].

A mobile robot platform with a fixed four-wheel arrangement chassis and an electronic system can be built around the Raspberry Pi and Arduino Uno Interfaces [13]. 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 can also be worked on [8]. In an unregulated and unpredictable environment, like city streets or wooded areas, the task becomes more challenging as the autonomous agent must be securely controlled to interact with other agents and avoid collisions while navigating [15] though a Fuzzy logic controller has been used for such navigation of a mobile robot-type tricycle in a partially known environment [12], it does not meet expectations in other cases. The most common approach involves employing various controllers such as P, PD, or PID controllers, which utilize real-time feedback from the observed system output, typically the robot's location, to determine the system input; however, this method requires precise models of the system, robot, and operating environment [7]. Artificial Neural Network (ANN) controllers can help address the rigour of precision by offering a more adaptive and data-driven solution. Instead of relying on explicit mathematical models of a system, ANN can learn complex mappings from inputs to outputs based on the available data.

3. Method

3.1. Modeling a Differential Drive Robot

A form of robot known as a differential drive robot has two wheels that are propelled by separate actuators. Combining the separate wheel movements results in the total movement of the robot. Each wheel on this kind of robot is managed by a different DC motor. The DC motor turns the wheels by producing torque from the voltage it receives as input. The position of the robot's body frame in relation to the world frame, as seen in Figure 1, is determined by the rotation of its wheels. A model shown in Figure 2 is created based on some presumptions to comprehend how the robot will move in response to a series of commands (known as forward kinematics) and to identify the commands necessary for the robot to achieve a desired movement (known as inverse kinematics).

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

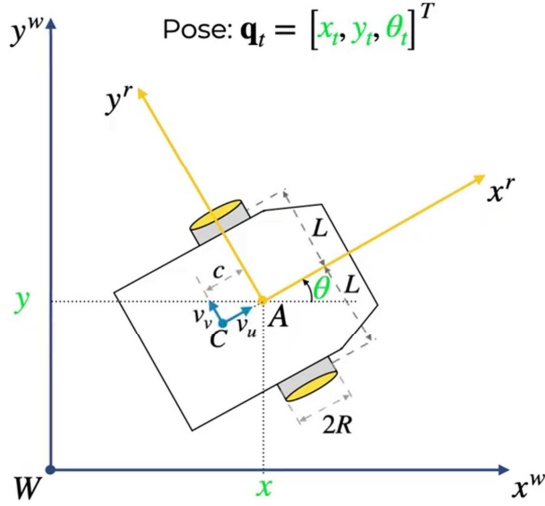


Figure 1. Differential drive robot [10].

As shown in [10]

Rigid body rotation:

$$\omega = [0, 0, \theta']^T \quad (1)$$

The velocity of point A in the body frame is given as

$$V_A^r = \begin{bmatrix} V_u \\ V_v - C\theta' \end{bmatrix} \quad (2)$$

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.

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

The rotational matrix is defined by the orientation angle θ .

3.2. General Kinematics Equations

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

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

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

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.

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

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:

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

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:

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

And equations (4) and (5) become:

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

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

With the no-skidding 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.

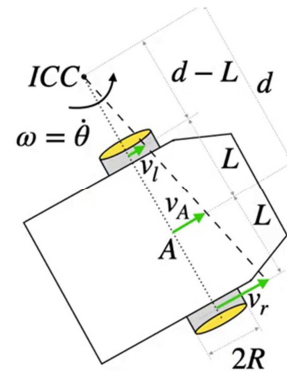


Figure 2. Instantaneous centre of curvature [10]).

$$d = L \frac{V_r + V_l}{V_r - V_l} \quad (12)$$

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

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.

3.3. Forward Kinematics

$$V_A = \frac{V_r + V_l}{2} \quad (14)$$

$$\omega = \frac{R}{2L} (\phi_r' - \phi_l') \quad (15)$$

$$\begin{bmatrix} V_A \\ \omega \end{bmatrix} = \frac{R}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \phi_r' \\ \phi_l' \end{bmatrix}$$

ϕ is the angular velocity of the wheel, ω is the angular

velocity of the robot. V_A is the linear velocity of the robot, and V_l and V_r are the linear velocities of robot wheels.

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

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

3.4. Inverse Kinematics

$$\begin{bmatrix} \phi'_r \\ \phi'_l \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & L \\ 1 & -L \end{bmatrix} \begin{bmatrix} V_A \\ \omega \end{bmatrix} \quad (17)$$

Relationship Between Voltage and Angular Velocity of the Wheel.

When voltage is applied to the terminals of DC motors, it results in the flow of current in the windings of the motor which generates a torque transmitted to the motor axis. The model of the motor is shown in Figure 3.

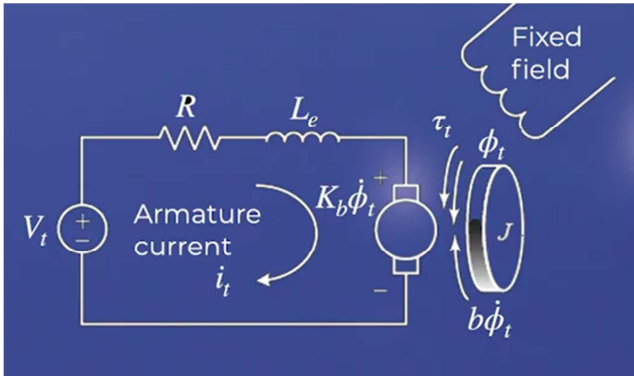


Figure 3. Model of robot wheel motor.

3.5. Electrical Subsystem of Robot Wheel Motor

$$V_t = Ri_t + L_e \frac{di_t}{dt} + K_b \dot{\phi}_t \quad (18)$$

Where $K_b \dot{\phi}_t$ is back emf.

$$\tau_t = K_i i_t \quad (19)$$

Where K_i is the torque constant.

3.6. Mechanical Subsystem of Robot Wheel Motor

$$J \phi''_t = \tau_t - b \dot{\phi}_t \quad (20)$$

Where $b \dot{\phi}_t$ is friction.

3.7. The Dynamics of the Motor of the Robot Wheel

$$\phi'_t = \left(R \frac{b}{K_i} + K_b \right)^{-1} V_t \quad (21)$$

The dynamics of the motor are faster than the dynamics of the robot, for this reason, a steady state behaviour is assumed in which the derivative terms in the electrical and mechanical subsystems are neglected so that there is an instantaneous relationship between the applied voltage and the wheel angular velocity. Based on this assumption equation (18) becomes equation (22) and equation (20) becomes equation (23).

$$V_t = Ri_t + K_b \phi'_t \quad (22)$$

$$J = \tau_t - b \phi'_t \quad (23)$$

4. Proportional Integral Derivative (PID) Controller Model for Navigation

The Proportional Integral Derivative controller is a three-in-one controller. The derivative controller helps to correct the steady state problem while the integral controller helps to correct the stability problem. In a Proportional Controller, the output is proportional to the error signal, in an Integral Controller, the output is proportional to the integral of the error signal and in a Derivative Controller, the output is proportional to the derivative of the error signal. In a combination of all three controllers, the output is given in equation (24), where $u(t)$ is the output in the time domain.

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt} \quad (24)$$

To get the transfer function of the PID controller, the Laplace transform of equation (24) is done, and it gives equation (25). Equation (26) is the transfer function of the controller. K_p , K_i , K_d are proportional, integral and derivative constants, respectively.

$$U(S) = E(s) \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (25)$$

$$\text{Transfer Function} = \frac{U(S)}{E(S)} = \left(K_p + \frac{K_i}{s} + K_d s \right) = \frac{K_p s^2 + K_i s + K_d s^3}{s} \quad (26)$$

From equation (26) it is seen that a pole is introduced at the origin. And there are two zeros. This means that one zero will compensate for the pole, and the other zero will increase the stability of the system by decreasing the steady state error. This suggests that if a system response varies abruptly in such a way that it is continuous as it changes direction, steady state error reduction may not be possible with the PID controller. That is, a situation of continuous overshoot may be encountered by the controller. To have a smooth control effect, one with the least overshoot, the PID controller is tuned for the optimum output of the controller for a system with a steady-state response. The Simulink block simulation of the PID-controlled robot for path navigation is shown in Figure 4.

5. Artificial Neural Network (ANN) Controller

Figure 5 is the Path Navigation Model with an ANN controller. Data (the transfer function of the PID controller) was extracted to the workspace of MATLAB from the PID control model during simulation, and the data was used to train the ANN controller. The idea was to have the ANN controller make better predictions with interpolation. The ANN is a Feedforward-Backpropagation network with the Levenberg-Marquardt Algorithm.

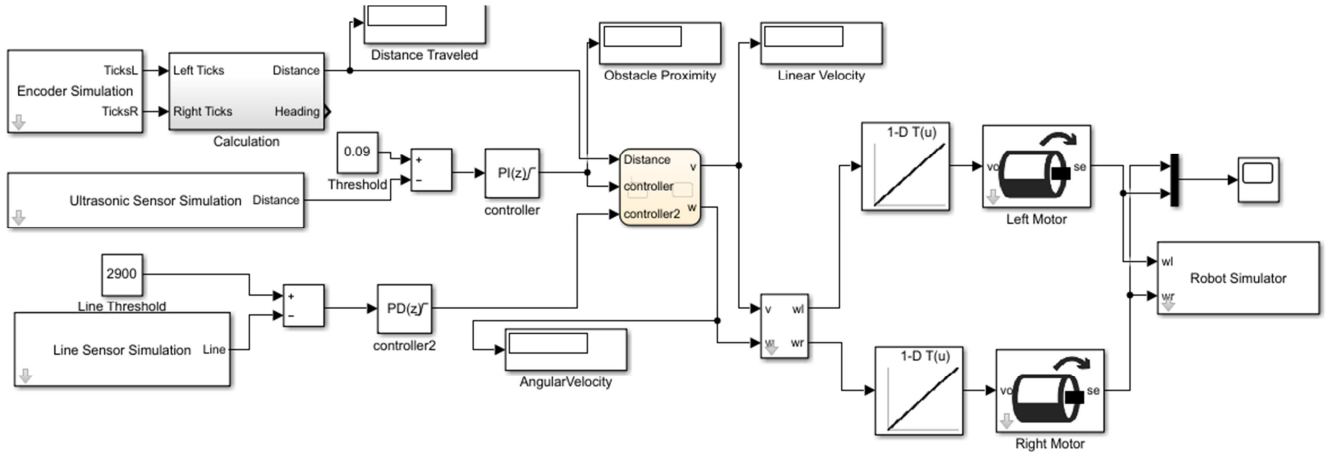


Figure 4. Simulink simulation of path navigation with a PID controller.

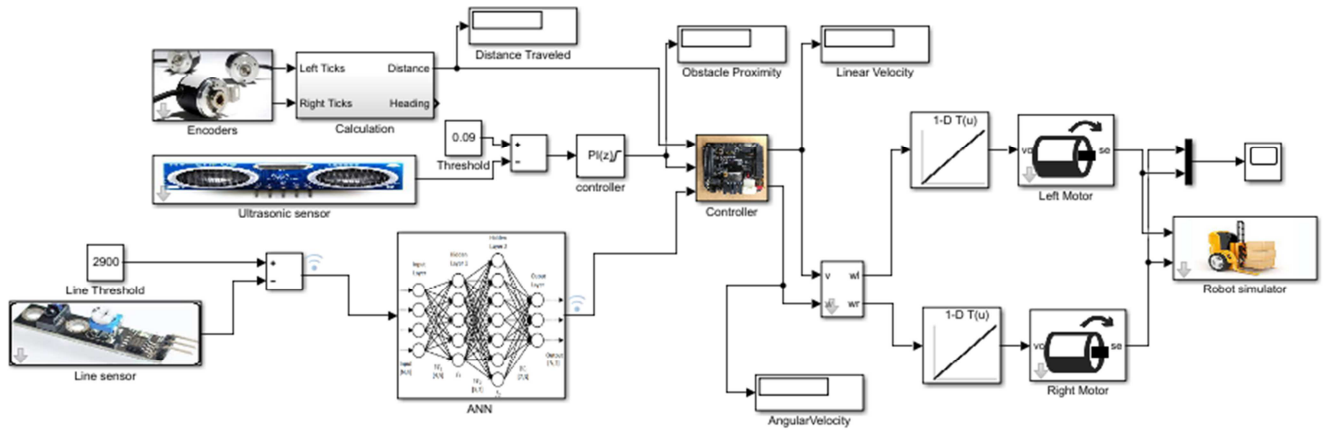


Figure 5. Simulink simulation of path navigation with an ANN controller.

6. Results and Discussion

Figure 6 is the signal output of the PID controller. As seen on the graph, there is a continuous overshoot in the response of the controller.

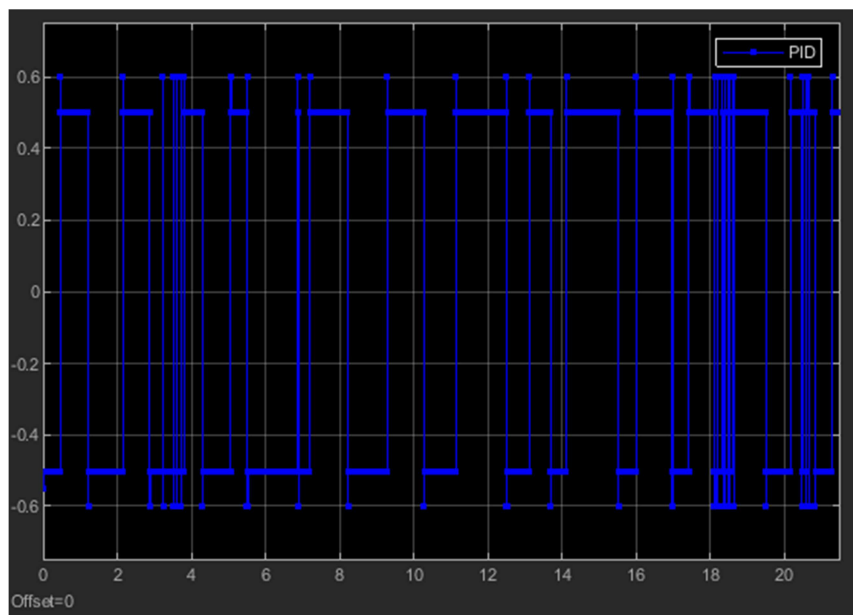


Figure 6. The signal output of the PID controller.

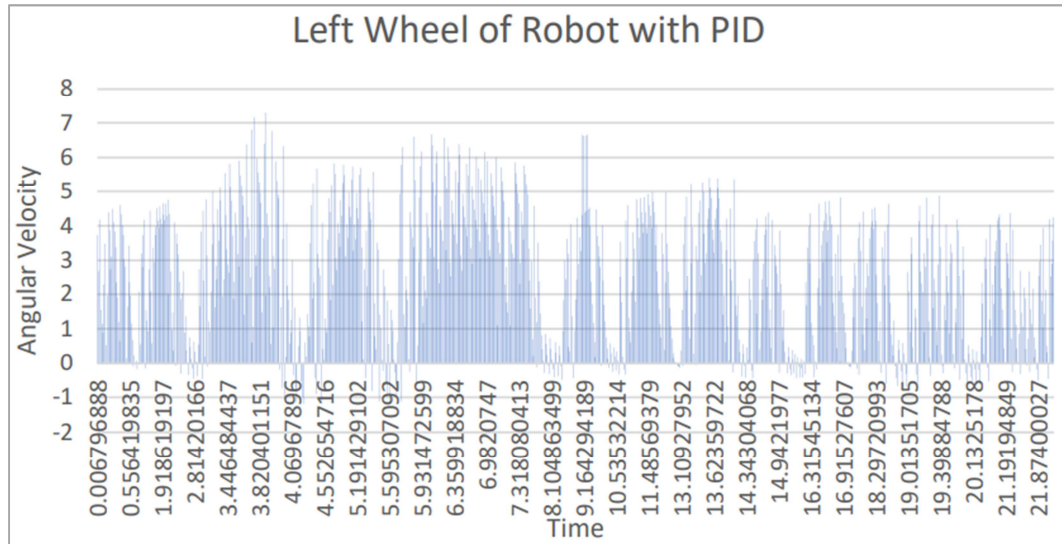


Figure 7. The behaviour of Robot Wheels with PID controller.

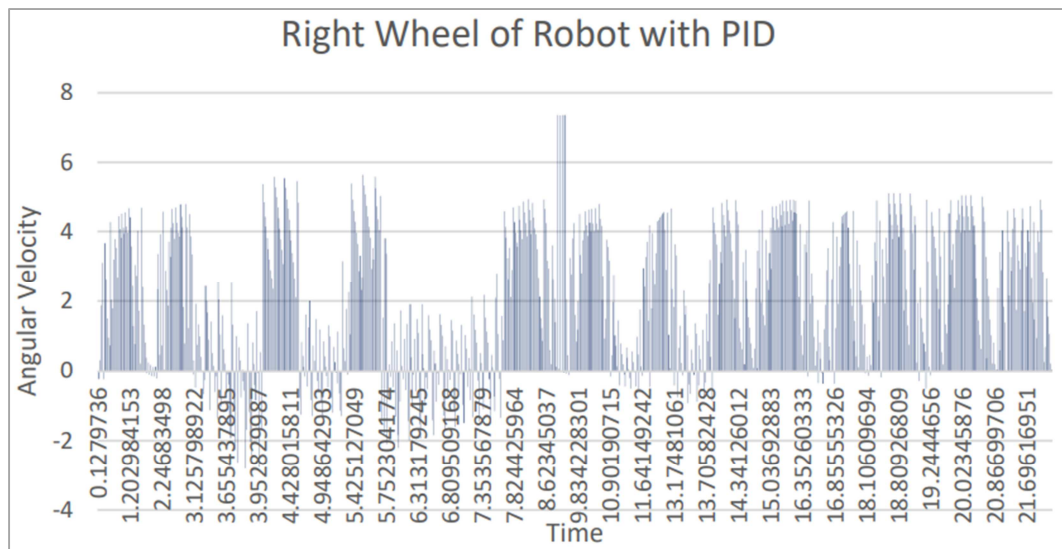


Figure 8. The behaviour of Robot Wheels with PID controller.

The graph in Figure 6 shows a pulse width modulation with the designed parameter of ± 0.5 units. As seen from the graph, there are impulses every time there is a change in polarity in the response of the control signal. The robot navigated through a curved line, and so, has its wheels turned in such a way as to keep it within the line as it moved. That is, it kept changing in angular velocity while the linear velocity was kept constant. The effect of such an impulse is seen on the graph of the robot wheels shown in Figures 7 and 8.

The error in Figure 6 is determined as follows:

Desired signal output = 0.5 units

Maximum signal output = 0.6 units

Error in signal output = 0.1 unit

The impulse given by the controller is 0.1 unit. This continuous impulse brings about shock in the system that results in power loss and also weakens the insulation resistance of the motor winding.

The percentage error is given as $\frac{0.1}{0.5} \times 100 = 20\%$

The effect of this was seen in the behaviour of the wheels. Ideally, the mean of the wheels should be the same, as both wheels should turn in the same direction at the same time, but the result of the graph showed a disparity in the mean. This is shown in Table 1.

The mean difference of PID-Controlled Wheels is $2.392 - 2.071 = 0.321$.

Percentage Mean Difference of PID Controlled Wheels: $0.321 \times 100 = 32.1\%$.

Both wheels receive signals from the wheel drivers simultaneously and are to have the same effect. The percentage mean difference shows that the system has a high skidding effect as the robot navigates on the line of the path. This means that the response of the controller on the robot wheels is low by 32.1%. That is, both wheels do not respond to the controller the same way. The result of the ANN-controlled robot showed that the overshoots had some contribution to this. Other factors that could contribute to this are the friction and

drag coefficients of the air.

Table 1. Data statistics for a motor wheel with a PID controller.

PID controlled wheels	Right motor wheel (rad/s)	Left motor wheel
Mean	2.392	2.071

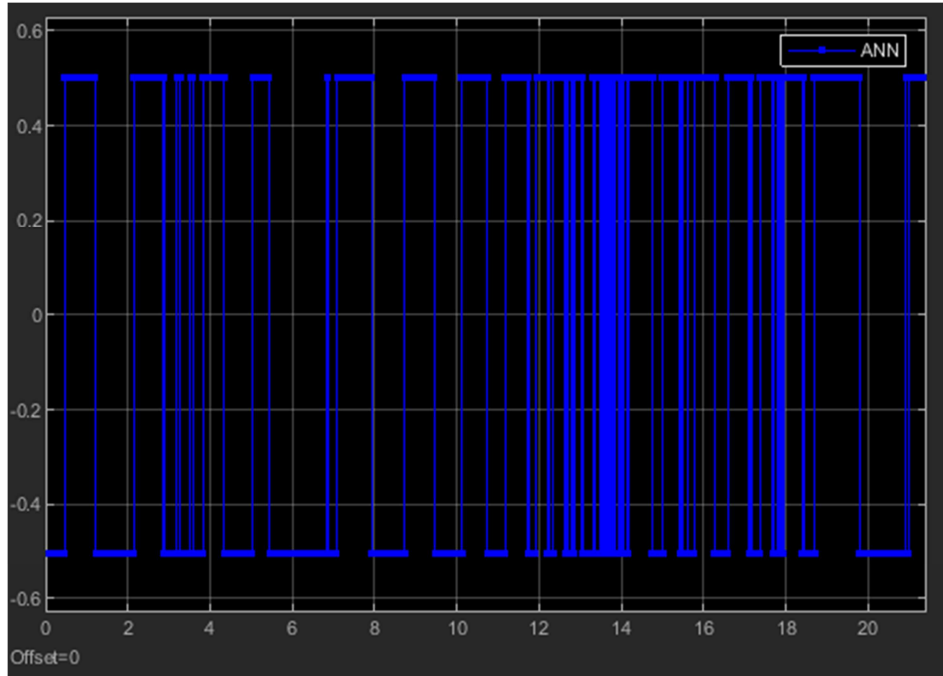


Figure 9. The signal output of an ANN controller.

Figure 9 is the response signal of the ANN controller when used to control the path navigation of the robot. A first observation is that the overshoots are not seen. They have become negligible. The effect of this is seen in the behaviour of the robot wheels, as seen in Figures 10 and 11.

After extracting data from the graph of Figure 9, it was seen that the maximum value of the signal is 0.5025 units. This gives an error of $0.5025 - 0.5 = 0.0025$.

The percentage error is $\frac{0.0025}{0.5} \times 100 = 0.5\%$

When this is compared with the response of the PID

controller which has a percentage error of 20%, it showed an improvement of $20\% - 0.5\%$ which is 19.5%. The effect of this is shown in the graphs of Figure 10 and Figure 11.

The response of the controller shows a drastic reduction in overshoot. The controller used interpolation to make predictions. It used backpropagation to reduce the error in the output signal. It is based on this understanding that the ANN controller was proposed in this project. The signal of the line sensor is 2900 ± 50 and it is designed to give a control signal response of ± 0.5 unit.

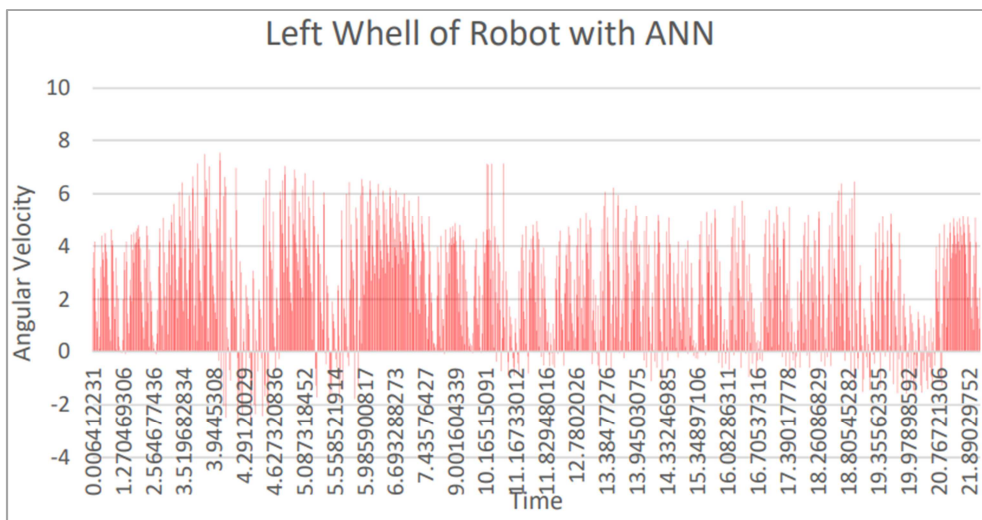


Figure 10. The behaviour of the Robot wheel with ANN controller.

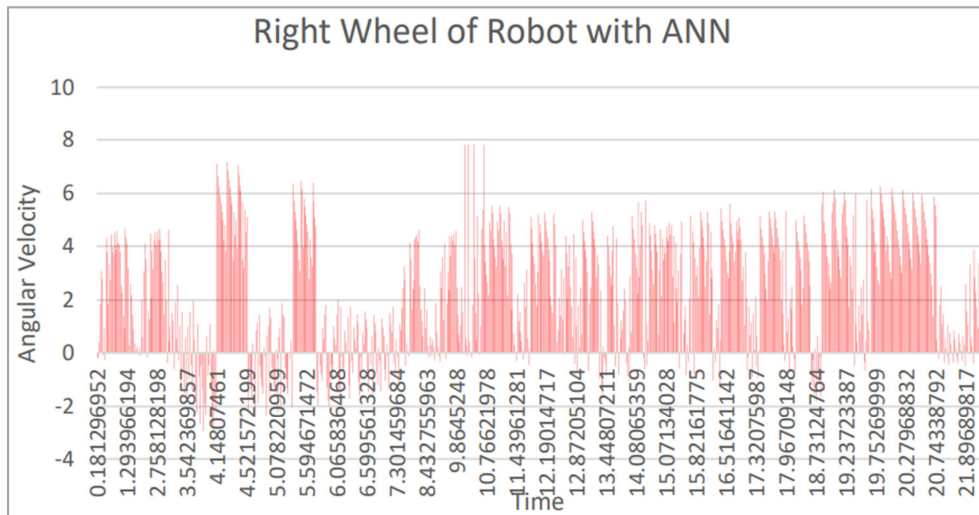


Figure 11. The behaviour of the Robot wheel with ANN controller.

Data for statistical analysis of Figure 10 and Figure 11 was extracted from the graph using MATLAB statistical tool. Table 2 gives the mean of the robot wheels.

Table 2. Data statistics for the motor wheel with an ANN controller.

ANN controlled wheels	Right motor wheel (rad/s)	Left motor wheel
Mean	2.326	2.136

The mean difference of ANN-Controlled Wheels is $2.326 - 2.136 = 0.19$

The percentage Mean difference of ANN-Controlled Wheels is $0.19 \times 100 = 19\%$

Performance improvement of wheels with ANN Controller: 32.1% (PID controlled wheel mean difference) – $19\% = 13.1\%$

From the result analysis, the ANN-controlled wheels have a 13.1% increase in response to the controller.

7. Conclusion

This research work on autonomous robot navigation using a differential drive robot has yielded significant advancements. By combining environment perception, ultrasonic obstacle detection, and line following for localization, the robot's navigational capabilities were enhanced. The integration of an artificial neural network (ANN) controller, trained with data from the PID controller, resulted in improved control performance, adaptability to changing conditions, reduced manual tuning, and increased control flexibility. These findings have profound implications for the industry, paving the way for intelligent and efficient autonomous robot navigation systems by enhancing controller performance with a trained network.

References

- [1] Aftf, M., Ayachi, R., Said, Y., Pissaloux, E., & Atri, M. (2019). Indoor Object Classification for Autonomous Navigation Assistance Based on Deep CNN Model. 2019 IEEE International Symposium on Measurements and Networking, M and N 2019 - Proceedings. <https://doi.org/10.1109/IWMN.2019.8805042>
- [2] Ajeil, F. H., Ibraheem, I. K., Azar, A. T., & Humaidi, A. J. (2020). Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment. June, 1–15. <https://doi.org/10.1177/1729881420929498>
- [3] Cheng, J., Cheng, H., Meng, M. Q. H., & 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>
- [4] Crestani, P. R., Von Zuben, F. J., & Figueiredo, M. F. (2002). A hierarchical neuro-fuzzy approach to autonomous navigation. Proceedings of the International Joint Conference on Neural Networks, 3, 2339–2344. <https://doi.org/10.1109/ijcnn.2002.1007507>
- [5] Emmi, L., Le Flécher, E., Cadenat, V., & Devy, M. (2021). A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture. Precision Agriculture, 22 (2), 524–549. <https://doi.org/10.1007/s11119-020-09773-9>
- [6] Eneh, P. C., Eneh, I. I., Egoigwe, S. V., & Ebere C. U. (2019). Deep Artificial Neural Network Based Obstacle Detection and Avoidance for a Non-Holonomic Mobile Robot. U., 16 (3), 1–14. <https://www.academia.edu/44028998/>
- [7] Engedy, I., & Horváth, G. (2009). Artificial neural network-based mobile robot navigation. WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings, 241–246. <https://doi.org/10.1109/WISP.2009.5286557>
- [8] Hank, M., & 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>

- [9] Harapanahalli, S., Mahony, N. O., Hernandez, G. V., Campbell, S., Riordan, D., & Walsh, J. (2019). Autonomous navigation of mobile robots in a factory environment. *Procedia Manufacturing*, 38 (2019), 1524–1531. <https://doi.org/10.1016/j.promfg.2020.01.134>
- [10] Innocent Ifeanyichukwu Eneh, Princewill Chigozie Ene, and Emmanuel C. Obasi, (2022) “Modelling a Deep Learning and Fuzzy Logic-Based Behavioral Approach for Autonomous Navigation of a Robot in a Global Positioning System (GPS) Denied Environment” *International Journal of Real-Time Application and Computing Systems (IJORTACS)*, Volume 1, Issue X, October 2022, pp. No. 21, pp. 270-282.
- [11] Khan, S., & Ahmmed, M. K. (2016). Where am I? Autonomous navigation system of a mobile robot in an unknown environment. 2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016, 56–61. <https://doi.org/10.1109/ICIEV.2016.7760188>
- [12] Melik, N., & 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>
- [13] 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>
- [14] Princewill Chigozie Ene (2022). The Kinematic Modeling of Four Mecanum Wheel Robot for Environment Mapping and Navigation. 2022 *International Journal of Real-Time Application and Computing Systems (IJORTACS)*, Volume 1, Issue X, October 2022, pp. No. 21, pp. 270-282. <http://www.ijortacs.com>
- [15] Zhilenkov, A. A., Chernyi, S. G., Sokolov, S. S., & Nyrkov, A. P. (2020). Intelligent autonomous navigation system for UAV in randomly changing environmental conditions. *Journal of Intelligent and Fuzzy Systems*, 38 (5), 6619–6625. <https://doi.org/10.3233/JIFS-179741>