International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Preliminary Analysis by using FCGA for Developing Low Power Neural Network Controller Autonomous Mobile Robot Navigation

M. Preetha[1], Archana A B[2], K. Ragavan[3], T. Kalaichelvi[4], M. Venkatesan[5]

Submitted: 18/10/2023 **Revised:** 07/12/2023 **Accepted:** 15/12/2023

Abstract: The desire for autonomous robots that can plan their paths and avoid obstacles is quickly increasing. In this study, we provide a machine - learning system design for robotic systems that detects and avoids obstacles using an artificial neural forecasting models and FPGA distributive processing algorithms. On the FPGA Virtex-II pro kit, the back propagation approach is used to implement learning and prediction. A floating point based computing approach is used to increase the neural network's flexibility and accuracy. The suggested paradigm is based on the notion of re-configurability, which lowers the cost and footprint of implementation. This suggested mobile robots robot design uses pipelined architecture that enhances predicting speed and lowers forecast latency. For emulation, the Xilinx 14.3 ISE simulation is utilised. The suggested methodology for managing the autonomous robot yielded good throughput and low power consumption in Place and Route outcomes.

Keywords: *Autonomous Mobile Robot, FPGA, Neural Network, Re-configurability, Path Planning and Obstacle Avoidance*

1. Introduction

Because of the precision of the work done by a robot, its use of robotic systems is fast expanding. The difficult problem for researchers is to achieve effective navigation, which includes navigation system and variable speed. Because of the programmable nature of the architecture, which helps to lower the cost of the design, the usage of Programmable Logic Array (FPGA) is expanding inside the area of mobile robots. Algorithms or different sophisticated computations, including such impediment sensing and artificial neural processing, can be accomplished utilising a complex FPGA hardware device. These FPGA systems specs allow for the incorporation of an artificial intelligence system for robot manipulators and collision avoidance. In most cases, robot manipulator navigating was based on the simple concept of detecting barriers. It must first detect the surroundings, then plan a moving path, then map the way, and lastly act on the mobile machine's planned course. The activities listed above are the most common tasks that every autonomous mobile robot is required to perform. About any robotic system, robot localization is a crucial phase in the implementation process and is accountable for the system's performance. We present a multilayer perceptron approach for effective autonomous mobile robot navigation and collision avoidance. Various hardware designs for fully independent autonomous navigation leveraging programmable microcontroller array to construct artificial neural networks and adjust parallelism of neural processing have been presented in the previous decade. This ANN approach is employed in a variety of applications, including video processing for high-speed data transfer in real-time applications, learning of large datasets, and high parallel processing performance.

To reach the goal while avoiding obstacles in a variety of surroundings, an animatronic drone must be capable of target localization, obstacle avoidance efficacy, accurate decision making, and action based on the choice. Currently, autonomous robots accomplish tasks with independence and cognition in real time. Classic techniques to this problem have lately been supplanted by new ones in order to accomplish the requisite performance. Different soft computing approaches, such as fuzzy logic, neural networks-based approach, adaptive resonance theory, and various hybrid intelligent systems, have been proposed to achieve this.

In recent years, scientists in the industrial automation have been drawn to neural network approaches for robot gps devices. N. M. Amosov [5] and R. Brooks [6] were the first to propose this strategy. In this paper, the authors suggest neural network modelling for a variety of scenarios, including dynamic, partly observable, and unknown scenarios of the environment. The navigation problem is primarily separated into the following tasks: Construction of the map, robot localization, path planning, and obstacle avoidance

Q. Gao et al. [9] present a fresh helpful advancement calculation based on a half-and-half refining estimate with both local and global inquiry abilities in this subject. To improve the managers' investigation and abuse properties, the strategy combines the worldwide pursue property of downpour woodlands calculation (RFA) with the rapid union of PSO. Furthermore, the suggested RFA-PSO efficiently solves a construction outline advancement problem derived as from step management of a snake-like robot.

The fractal neural nets (WNNs) structure, on the other hand, achieves better identifiable proof execution by combining wave principles and manufactured neuronal pathways (ANNs) methodologies. Furthermore, nonlinear frameworks were recognised and controlled using repeated wavelet neuronal pathways (RWNNs), which combine qualities such as the dynamic responsiveness to interrupted neuronal pathways (RNNs) and the rapid meeting of WNNs. The use of Meets certain standards and RWNNs also isn't covered in this article; nonetheless, effective execution of the architectures necessitates use of several levels with many neurons.

[1]*Professor & Head, Department of CSE,*
Prince Shri Venkateshwara Padmavathy Engineering College,
Email: smpreetha14@gmail.com
ORCID: 0000-0001-8483-9871
[2]*Assistant Professor in CSE-Cyber Security Department,*
Madanappalli Institute of Technology & Science
Email: archanaab@mits.ac.in
ORCID: 0009-0005-1448-9438
[3]*Assistant Professor Senior Grade I,*
School of Computer Science and Engineering,
Vellore Institute of Technology, VIT University, Vellore -632014
Tamilnadu, India
Email: ragavan.k@vit.ac.in
ORCID: 0000-0002-0833-5396
[4]*Professor, Dept of Artificial Intelligence and Data Science*
Panimalar Engineering College
Email: kalaichelvi2012@gmail.com
ORCID: 0009-0001-9999-6965s
[5]*Professor, ECE Department,*
PBR Visvodaya Institute of Technology and Science, Kavali, AP
Email: hellomvenkat@gmail.com
ORCID: 0009-0002-1736-6220

2. Proposed Model of Artificial Neural Network

Convolutional neural networks are a sort of artificial intelligence that is inspired by and guided by the functions of the human brain. ANNs are made up of massively parallel, highly linked handling components. Every operation processing, or neuron, is theoretically too myopic to learn anything significant on its own. The zenith of many cells inside a neo cortex is the source of notable training limit and, as a result, handles management. The learning ability of ANNs has been demonstrated in a variety of applications, including design recognition, capacity estimation/forecasting, and robot control. ANNs are divided into two categories based on how they learn: (a) supervised learning and (b) unsupervised learning. We employ a supervised learning technique with a back propagation algorithm that uses inputs and projected outputs for the suggested model of robotic gps module.

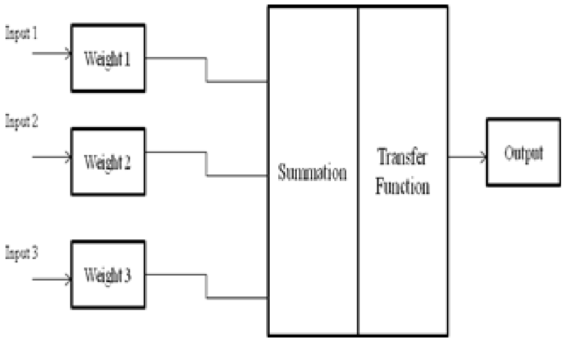

Fig. 1. Proposed Perceptron ANN

The suggested back propagation technique for robot navigation is depicted in Figure 2. An ANN that uses the back propagation technique includes five phases of operation, as per Rumelhart et al. [16].

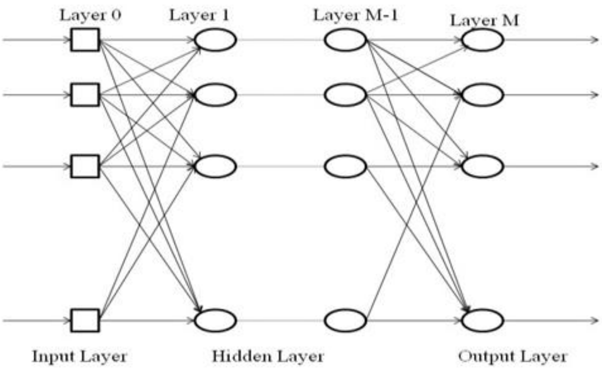

Fig. 2. Multi layer Perceptron Model

We use backward calculation in the deep residual method to optimize the parameters in the network training, which helps to decrease the error between the target and actual output. The following is the definition of this method:
Begin with the network's output layer and work your way input to the output plane to calculate the local gradients. A graze network is used to accomplish a forward computation, in which material from the bottom layer of said human brain is supplied to the network's higher layer. It may also be seen in Fig. 2, where other neurons are labelled as 1, 2, 3, 4, 5, and 6.
The formula for the calculation is as follows:

$$H_k^{(s)} = \sum_{j=1}^{N_{s-1}} w_{kj}^{(s)} O_j^{(s-1)} + \theta_k^{(s)} \quad (1)$$

Where, J$< k$ and s = 1; ,:,:,: ,M

$H_k^{(s)}$= Weight sum of the Kth neuron is Sth layer

$W_{kj}^{(s)}$ = corresponding weight from jth unit

$O_j^{(s-1)}$ = neuron output of jth neuron

$\theta_k^{(s)}$ = bias of kth neuron

$$\theta_k^{(s)} = f\left(H_K^{(S)}\right) \quad (2)$$

$f\left(H_K^{(S)}\right)$= activation function based on the weighted sum
By using these equations, the non-linear activation function is computed as:

$$f(x)_{logsig} = \frac{1}{1+\exp(-x)} \quad (3)$$

The computation method is shown in Eq. 4 to Eq. 6
Computation of weight by using Eq. 7
Update all the weights by using Eq. 8

$$\mathcal{E}_k^{(s)} = \begin{cases} t_{k-}O_k^{(s)} & s = M \\ \sum_{j=1}^{N_{s+1}} w_{kj}^{(s+1)} \sigma_j^{(s+1)} & s = 1, \dots. M-1 \end{cases} \quad (4)$$

Where,

$\mathcal{E}_k^{(s)}$ = error value for neuron

$\sigma_j^{(s+1)}$ = local gradient for neurons

$$\sigma_j^{(s)} = \mathcal{E}_k^{(s)} f'\left(H_k^{(s)}\right), s = 1, \dots . M \quad (5)$$

Where, $f'\left(H_k^{(s)}\right)$ is the derivative of activation function of the network, which can be computed as partial derivationof the active function.

$$f'\left(H_k^{(s)}\right) = \frac{\delta\left(\alpha_k^{(S)}\right)}{\delta\left(H_k^{(S)}\right)} = \left(1-\alpha_k^{(s)}\right)\alpha_k^{(s)} \quad (6)$$

$$\Delta w_{kj}^{(s)} = \eta \delta_k^{(s)} O_j^{(s-1)} \; k = 1, \dots N_s \quad (7)$$

$\Delta w_{kj}^{(s)}$ is the variation in the weight, which is defined as:

$$\Delta w_{kj}^{(s)}(n+1) = \Delta w_{kj}^{(s)}(n) + \Delta w_{kj}^{(s)}(n) \quad (8)$$

3. Implementation

The neural model was written in VHDL and run. The VHDL language combines the VHSIC (Ultra High Speed Integration Circuits) and Density lipoprotein (Hardware Description Language) programming languages. The simplest way to deal with the situation is to use a top-down technique, which entails breaking down a complex outline into simpler plans or modules, with each module being s actually with more prominent topics of focus or divided into more subsystems. In a synthetic neuron with only three dendrites as inputs, we used a framework-based construction. Around 50 activate a neuron. The ability to learn is one of the most important properties of the brain systems. Training is the modification of the actuated behavior for cooperation in neurons. That data is being spoken to during the strengths of the connections between the neurons in simulated neuronal pathways.
The learning process predicts that these connections will alter in little amounts. The neural system learns by altering the system's weight calculations. Weights are important because they are associated to neurotransmitters and can increase or decrease the indications that are connected to a neurological link. As previously stated, we have defined three adjustable values as sources, each of which is coupled to a dendrite to improve learning capacity (Fig. 3).

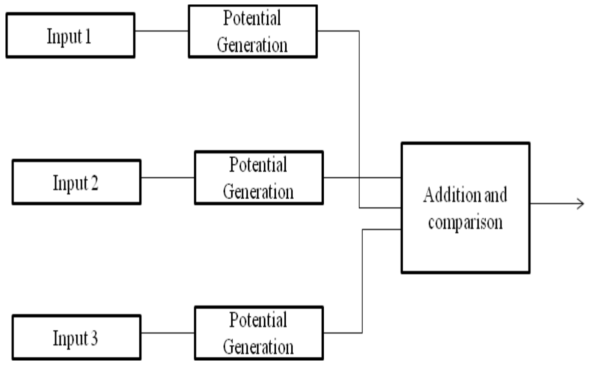

Fig. 3. Model for input neuron comparison

Every dendrite is connected to each weight. For render this system asynchronous, a clock indicator is required. When the neuron model receives the inputs, it consists of two components: (a) action production based on the input, and (b) time threshold assessment. This is how it's defined: Figure 4 shows the creation for prospective inputs and a reliability evaluation.

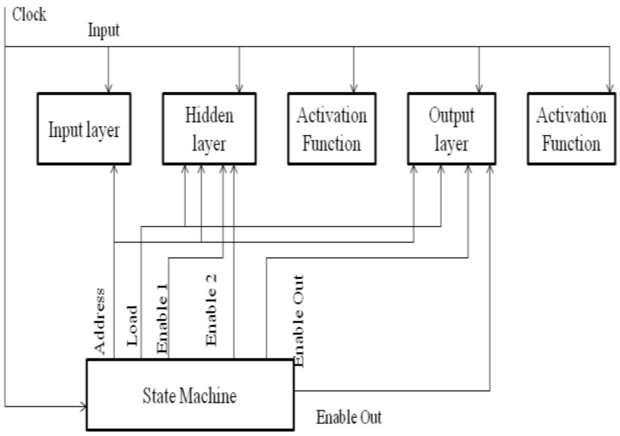

Fig. 4. Block Diagram of ANN

4. Results and Discussion

The outcomes of the suggested model are discussed in this section. The Virtex-II Pro kit from Xilinx was used to create this model. Table 1 shows how neurons' resources, state machines, and activation functions are used. Similarly, Table 2 shows how much of the design's resources were used.

Table 1. Utilization of Resource

	Neurons	**State Machine**	**Activation Function**
Slices	14	14	-
Slice Flip-Flops	23	18	-
4-Input LUTs	23	18	-
Block RAMs	-	-	2

Table 2. Synthesis Results of the Proposed Scheme

	Used	**Available**
Number of Slices	386	63168
Number of Slice Flip-Flops	329	126336
Number of 4-Input LUTs	876	126336
Number of bonded IOBs	37	768
Frequency	383.3 MHz	

System failures in hardware processor-based applications cause the mechanism to stop working. The main cause is that the CPU architecture is expensive. Studies have discussed a neural network with several units and effective high availability. Concurrent design and distributed computing, in addition to these benefits, boost performance. Due to the obvious speed & low power of FPGA-based mobile robots, they are widely employed for a wide range of applications.

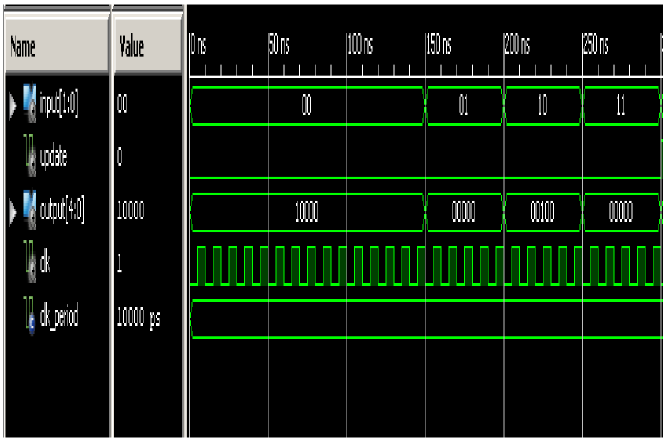

Fig. 5. Input of Neural network

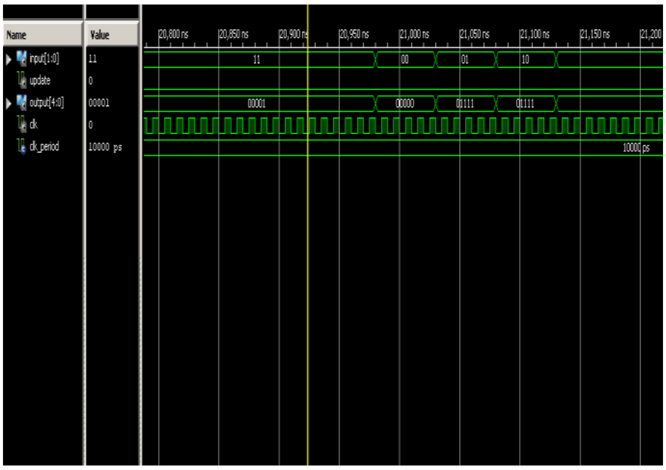

Fig. 6. Output of Neural network

Table 3. Power Consumption Results of the Proposed Scheme

Power Summary		
Total Power	Dynamic Power	Static Power
1152 mW	7.1mW	133 mW

The goal of this research is to offer a hardware method for implementing RBF Artificial Neural Network (Radial Basis Function type Neural Network) circuits in FPGA (Field Programmable Gate Array) circuits for mobile robot navigation, which comprises robot route planning and collision avoidance. Throughout this research, an algorithm is designed, and the architecture for mobile robot navigation is discussed, as well as the processing units for the robot design.

5. Conclusion

This paper presents an independent core tech for the mobile robots controller, and a gear implementation of a NN algorithm. The system was built using a cascaded electronics technique. When compared to an usual programming use of NN in Matlab, the design programmed in VHDL was run on a Xilinx Virtex-II Pro FPGA and performed well on two common benchmark challenges. The platform is intended for applications that require a lot of NN processing time. The framework enables graze Reinforcement learning to be implemented on FPGAs, resulting in increased performance. The NN equipment is built using an FPGA-based method, which makes it easier for experts to design to a particular application. Since all of the robot's functionality is kept inside the register, the suggested system requires less amount of memory it. This boosts the suggested model's computation time.

References

[1] Chakravarthy, N. and Jizhong Xiao, "FPGA-based Control System for Miniature Robots," International Conference on Intelligent Robots and Systems 2006 IEEE/RSJ, pp. 3399-3404, 9-15 Oct. 2006.

[2] Guanghua Zong, Luhua Deng and Wei Wang, "A Method for Robustness Improvement of Robot Obstacle Avoidance Algorithm," IEEE International Conference on Robotics and Biomimetics, ROBIO-06, pp. 115-119, 17-20 Dec. 2006.

[3] Ziemke, T, "Remembering How to Behave-Recurrent Neural Networks for Adaptive Robot Behavior", in Recurrent Neural Networks: Design and Applications, CRC Press 2000. ISBN 0849371813. pp. 355–390.

[4] Laboratory of Intelligent Systems, Ecole Polytechnique Fdrale de Lausanne, Switzerland [online]. [quoted 2008-08-21].

[5] Amosov, N. M.., Kussul, E. M., Fomenko and V. D.: "Transport Robot with a Neural Network Control System", Advance papers of the Fourth Intern Joint Conference on Artificial intelligence, pp. 1-10, 1975.

[6] Brooks R.., "A Robust System Layered Control System for a Mobile Robot" IEEE Trans. on robotics and automation RA-2, 14-23, 1986.

[7] Janglova, D, "Neural Networks in Mobile Robot Motion", in International Journal of Advanced Robotic Systems, 1(1) (2004) 15-22.

[8] W. de la Torre, F. Jurado, M. A. Llama, and R. Garcia-Hernandez, "Takagi-Sugeno fuzzy dynamic regulator for a pendulum on a cart system," in Proceedings of the 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE '13), pp. 52–57, Mexico City, Mexico, October 2013.

[9] Qin Gao, Zhelong Wang and Hongyi Li, "An Optimization Algorithm with Novel RFA-PSO Cooperative Evolution: Applications to Parameter Decision of a Snake Robot".

[10] Y. Alanis, M. Lopez-Franco, N. Arana-Daniel, and C. LopezFranco, "Discrete-time neural control for electrically driven nonholonomic mobile robots," International Journal of Adaptive Control and Signal Processing, vol. 26, no. 7, pp. 630–644, 2012.

[11] Farmahini-Farahani, S. M. Fakhraie, and S. Safari, "SOPC-based architecture for discrete particle swarm optimization," in Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on, Marrakech, Dec. 2007, pp. 1003–1006.

[12] D.E. Rumelhart, G .E. Hinton and R. J. Williams, "learning internal representations by error propagation", Parallel Distributed Processing, Vol. I. pp.312-362, MIT press. (1986)

[13] Chaomin Luo, Jiyong Gao, Xinde Li and Hongwei Mo; Qimi Jiang, "Sensor-based autonomous robot navigation under unknown environments with grid map representation," in Swarm Intelligence (SIS), 2014 IEEE Symposium on Swarm Intelligence, pp.1-7, 9-12 Dec. 2014.

[14] Chaomin Luo, Yang, S.X., Hongwei Mo and Xinde Li, "Safety aware robot coverage motion planning with virtual-obstacle-based navigation," in Information and Automation, 2015 IEEE International Conference on Information & Automation, pp.2110-2115, 8-10 Aug. 2015.

[15] Najmuddin Aamer and S. Ramachandran, "Neural Networks Based Adaptive Approach for Path Planning and Obstacle Avoidance for Autonomous Mobile Robot (AMR)" International Journal of Research in Computer Applications and Robotics (IJRCAR), Vol.3 Issue 12, Pg.: 66-79, December – 2015.