

# Pipelined, High Speed, Low Power Neural Network Controller for Autonomous Mobile Robot Navigation Using FPGA

Najmuddin Aamer, S. Ramachandran

**Abstract** - The demand for autonomous robots which incorporates efficient path planning and obstacle avoidance is increasing rapidly. In this paper, we have proposed a neural network based hardware architecture for autonomous mobile robot which is able to detect and avoid obstacles by using prediction model of neural network and distribution computation techniques using FPGA. Learning and prediction is implemented by using back propagation method on FPGA Virtex-II pro kit. For flexibility and accuracy of the neural network, floating point based computation method is applied. The proposed model uses the principle of reconfigurability which reduces the implementation cost and area. In this proposed architecture of autonomous mobile robot, pipelined architecture is used which increases the speed and reduces the delay for the prediction. Simulation is performed by using Xilinx 14.3 ISE simulator. Place and Route results exhibit high throughput and low power consumption achieved using this proposed model for controlling the autonomous robot.

**Index Terms**— Autonomous Mobile Robot, FPGA, Neural Network, Pipeline, Reconfigurability, Path Planning and Obstacle Avoidance.

## I. INTRODUCTION

Use of autonomous mobile robot is increasing rapidly due to the preciseness of the task performed by a robot. For researchers, the challenging task is in accomplishing efficient navigation, namely, obstacle avoidance and speed control. In the field of mobile robot navigation use of Field Programmable Gate Array (FPGA) is increasing because of the reconfigurable nature of the architecture which helps to reduce the cost of the design. By using FPGA, the algorithms or various complex calculations can be performed using complex FPGA hardware system such as obstacle sensing, neural network computation etc. These specifications of FPGA system provide the advantage to incorporate the artificial system of intelligence for mobile robot navigation and obstacle avoidance system. Generally mobile robot navigation is based on the simple idea of sensing the obstacles. It need to sense the environment, then the planning of the path for the movement, next is mapping then finally act upon the planned path of the mobile robot.

Autonomous robot working is defined as follows:

- (a) Collection of information from the surroundings,
- (b) Travelling to the destination without any human assistance,
- (c) Avoidance of the obstacles, and
- (d) Acting according to the situations of the environment.

The above mentioned tasks are the main tasks which are expected from any autonomous mobile robot. For any autonomous mobile robot, localization of robot is an initial step which is critical for implementation and responsible for the performance of the system. For efficient navigation of autonomous mobile robot and obstacle avoidance, we use artificial neural network system as our proposed model.

In the last decade, various hardware architectures have been proposed for autonomous mobile robot navigation using reconfigurable field programmable array to realize the artificial neural network and to adapt parallelism of the neural processing. This method of ANN is used for various applications such as video compression for high rate of data transmission for real time applications, learning of huge datasets and high efficiency of parallel processing. The efficiency of the neural network in hardware applications can be as follows:

**Speed:** Hardware implementation of neural network is a specialized hardware which offers high computational power and increased speed at the low price of manufacturing. It takes advantage of neural network's parallelism process and distributed computing.

**Cost:** Due the processing specification of the neural network, it uses less number of components to build the hardware and helps to decrease the power consumption for the robot navigation.

**Hardware performance Up-gradation:** In the hardware processor based applications, system stops functioning due to the faults in the system. Main reason is inefficiency in the processor architecture. Neural network is proposed by researchers which contains multiple cores and efficient fault tolerance. Along with these advantages, the parallel architecture and distributed computing offers performance improvement.

FPGA based mobile robots are widely used for various applications because of the performance and low power consumption with reconfigurable architecture. FPGA based system to control the hardware and software has been studied in Ref. [1]. For autonomous robot navigation, path planning is one of the important elements. Finding a feasible path for the robot navigation is presented in Ref. [2] which is developed using Matlab GUI. For robot navigation, two main conditions are defined as obstacle avoidance and detection of obstacles.

Manuscript published on 30 January 2016.

\*Correspondence Author(s)

**Najmuddin Aamer**, Research Scholar, Department of Electronics & Communication Engineering, Vinayaka Mission University, Salem, Tamil Nadu, India.

**S. Ramachandran**, Department of Electronics & Communication Engineering, SJB Institute of Technology, Bangalore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

In this work, we focus on the motion planning and obstacle avoidance problem in the field of robot navigation which is the key point in the robotics. According to this method, robot is given initial positions and angles, target locations and angles in the defined workspace. In this workspace, the autonomous robot provides the continuous path from the initial location and coordinates points to achieve the target points or destination with the capacity of avoiding the obstacles. In this procedure, the path planning is divided into two sub problems which are finding the space and finding the path. For a similar problem of path planning and obstacle avoidance, Guanghua Zong et al. had presented a new method based on the sonar rings and discussed the drawbacks of the method [2]. To achieve the destination with obstacle avoidance in the various environments, the autonomous robot must have the capability of localization of the target, efficiency in the obstacle avoidance, proper decision making and performing action according to the decision. Currently autonomous robots are performed in real time with autonomy and intelligence. To achieve the required performance, recently classical approaches for this issue have been replaced by new ones. In order to achieve this, various soft computing methodologies have been proposed which are fuzzy logic, neural networks based approach, adaptive resonance theory and various hybrid intelligent systems.

Utilization of repetitive neural systems in autonomous robot navigation control was acknowledged by Ziemke [3]. The author assessed four architectures of intermittent neural system, which he actualized in the control arrangement of a smaller than expected robot named Khepera.

Researchers at the Laboratory of Intelligent Systems in Switzerland [4] have been creating wise route frameworks for self-ruling portable robots for quite a while. The center of their route frameworks comprises neural systems, which adjust to the scene environment through a hereditary calculation utilizing self-ruling collaboration of the framework with its surroundings [2].

The objective of this paper is to present a hardware solution for implementation in FPGA (Field Programmable Gate Array) circuit of RBF ANN (Radial Basis Function type Artificial Neural Network) used for mobile robot navigation which includes robot path planning and obstacle avoidance. In this paper, an algorithm is proposed, the architecture used for the mobile robot navigation will be presented and the technology used for the robot architecture will be described. The rest of the paper is organized as follows. Related work in this manuscript is explained in Section II. Section III describes the proposed simulator. Experimental results are presented in Section IV to show the system performance. Finally, Section V presents a brief conclusion.

## a. Related Work

In this section, we discuss about the related work in the field of the autonomous robot navigation system. In recent years neural network approach for robot navigation system has attracted researchers in the field of robotics. This method was introduced by the N. M. Amosov [5] and R. Brooks [6]. In this work, present authors have proposed the modeling of neural network for various scenarios, namely, dynamic, partial observable and uncertain scenarios of environment. Mainly the problem of navigation is divided into following tasks: Building the map, Localization of the robot, planning the path and avoiding the obstacles.

In various researches, multi-layer perceptrons (MLP) with back propagation learning method is used for the navigation of the robot. Janglova et al. [7] presented a method to overcome the drawbacks of the MLP neural network by utilizing hybrid model for navigation. This hybrid model consists of principle component based analysis for preprocessing and neural networks. Obtained results from this model show the performance of the model in terms of reduced learning time.

M. A. Llama et al. [8] presents a configuration of fuzzy controllers for an under actuated framework. In light of the fuzzy models, three Takagi-Sugeno fuzzy controllers are proposed by means of Parallel Distributed Compensation. A vital development in the field of evaluative calculations has happened as of late, where Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been utilized as a part of an incredible assortment of utilizations with high achievement. Both techniques have a few disservices: GA has moderate union and PSO may be gotten in the trap of neighborhood streamlining. To keep away from those detriments, approaches that consolidate these two calculations have been proposed.

In this theme, a novel helpful advancement calculation is displayed by Q. Gao et al. [9], which depends on a half and half streamlining calculation with both nearby and worldwide inquiry abilities. The strategy consolidates the worldwide pursuit property of downpour woods calculation (RFA) and the quick union of PSO to enhance the investigation and abuse property of the operators. Besides, a building outline advancement issue taken from the step control of a snake-like robot is effectively actualized by the proposed RFA-PSO.

In Ref. [10] by C. López-Franco et al., the authors proposed a converse ideal neural controller for a nonholonomic versatile robot with parameter vulnerabilities and obscure outside aggravations. Here, a discrete-time repetitive high-request neural system (RHONN) is prepared with a broadened Kalman channel with visual input, alongside ideal control hypothesis. This system is utilized to decide the control flags that will compel a procedure to fulfill physical limitations and minimize an execution foundation all the while. Ordinarily, in ideal control hypothesis, a cost practical is characterized as capacity of the state and the control variables; sadly, this system requires illuminating the Hamilton-Jacobi-Bellman (HTB) mathematical statement, which is not a simple errand. Considering this, C. López-Franco et al., utilized converse ideal control to evade the arrangement of the HTB mathematical statement; for sure, the information of the opposite ideal control is dictated by visual criticism.

Then again, the wavelet neural systems (WNNs) structure, which has emerged from joining wavelets ideas with manufactured neural systems (ANNs) approaches, accomplishes better recognizable proof execution. Moreover, repetitive wavelet neural systems (RWNNs), which join properties, for example, dynamic reaction of intermittent neural systems (RNNs) and the quick meeting of WNNs have been utilized to recognize and to control nonlinear frameworks. The utilization of WNNs and RWNNs is not another subject in the writing; be that as it may, the execution of these structures includes the utilization of various layers with numerous neurons.

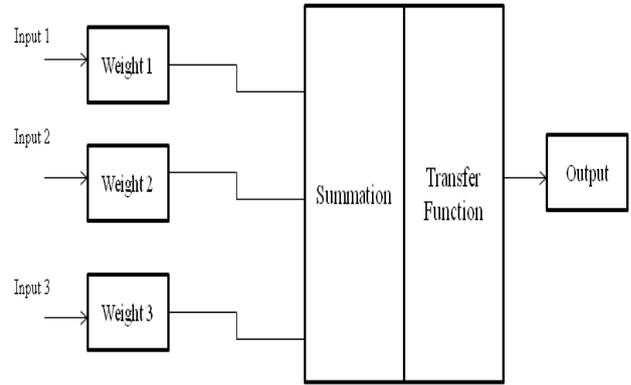
Considering this, L. A. Vázquez et al. present a novel article in Ref. [11]. A consistent time decentralized wavelet neural distinguishing proof and control plan taking into account the structure of an intermittent high-request neural systems (RHONN) model with wavelet Morlet enactment capacity has been proposed. The subsequent structure has been named repetitive wavelet first-request neural system (RWFONN). Moreover, the configuration of the proposed controller utilizes a back stepping approach. The preparation for the RWFONN is performed web utilizing the separated blunder (FE) calculation and another neural structure of a solitary layer and a solitary neuron. The execution of the proposed plan for direction following is approved by means of exploratory results performed in a robot controller of two DOF with obscure parameters.

The present authors in a previous work [12] have proposed a new method for the autonomous robot navigation by using Matlab tool. This work focuses on the path planning and obstacle avoidance modelling for the robot. Proposed model considers dynamic obstacles environment also to show the obstacle avoidance efficiency of the model. Similarly in another work [13], we proposed a model for autonomous robot navigation to overcome the drawbacks of the existing model of the neural networks. For path planning and obstacle avoidance, kinematic based model was introduced in this work. PASSAVOID [14] is a navigation scheme, which avoids braking inevitable collision states (i. e., states that regardless of the robot’s trajectory lead to a collision) to achieve safety in the presence of moving obstacles. Since PASSAVOID is designed to operate in completely unknown environments, it ensures that the robot is at rest when a collision occurs (passive safety). The motion dynamics of the robot have only been considered in simulation. Farmahini-Farahani et al. have implemented PSO within a system on a programmable chip framework [15]. The authors utilized a hardware implementation of the discrete version of PSO. A soft-core Altera NIOS II embedded processor was used to compute the fitness function and implemented on an Altera Stratix 1S10ES Development Kit. Performance was sacrificed in exchange for flexibility by implementing the fitness function in software. Sensor based navigation approach and map generation based navigation of autonomous robot was proposed by Chaomin Luo [17-18].

**b. PROPOSED MODEL**

**Artificial Neural Network**

Artificial neural networks are the type of artificial intelligence, which have been designed according to, and propelled by the procedures of the human brain. Fundamentally, ANNs comprise enormously parallel, profoundly interconnected handling components. In principle, every processing component, or neuron, is very shortsighted to learn anything important all alone. Noteworthy learning limit, and consequently, handling control just originates from the zenith of numerous neurons inside a neural system. The learning capability of ANNs has been shown in diverse territories of use, for example, design acknowledgment, capacity estimation/forecast, and robot control.



**Figure 1. The Perceptron Model**

**c. Backpropagation Algorithm**

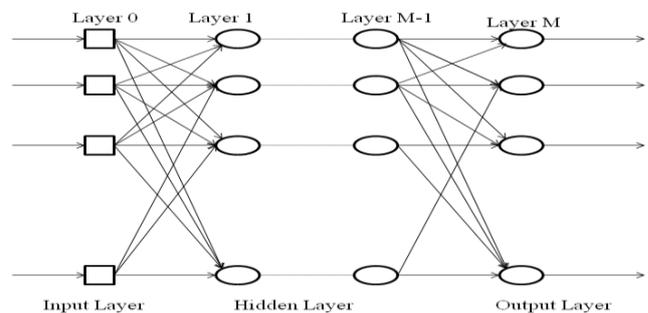
ANN is classified into two types based on their learning procedure: (a) supervised learning (b) unsupervised learning. For the proposed model of robot navigation system we use supervised learning procedure by using back propagation algorithm which uses inputs and predicted outputs.

Fig. 2 gives the overview of the proposed back propagation algorithm for robot navigation. According to Rumelhart et al. [16], an ANN using the back propagation algorithm has five steps of execution.

**d. Initialization of Neural network**

According to the neural network initialization process, the following parameters to be determined by the training process:

1. Computation of the weight of layers, where weight is given by  $w^{(s)}_{kj}(n)$  and  $n$  is the neuron of the particular layer of the network in  $j^{th}$  unit in  $(s - 1)^{th}$  layer. The computation of the weights is performed during the  $n^{th}$  iteration of back propagation, where  $n = 0$ .
2. Computation of the learning rate expressed as  $\eta$ . This is a scaling factor which is used for determining the size of the steps during the back propagation. The range of this constant is 0.1 to 0.5.
3. Biasing of the neuron  $\theta_k^s$  is similar to the weight of the layer.



**Figure 2. Multilayer Perceptron**

**e. Forward Computation**

To perform the forward computation, a feed-forward network is utilized in which data from the lower layer of the neural network is given to the upper layer of the network. It can be seen from Fig. 2, where the neurons are represented as 1 ... N. The computation is expressed 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  $s = 1; \dots; M$

$H_k^{(s)}$  = Weight sum of the  $k^{th}$  neuron in  $s^{th}$  layer

$w_{kj}^{(s)}$  = corresponding weight from  $j$ th unit

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

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

$$o_k^{(s)} = f(H_k^{(s)}) \quad (2)$$

$f(H_k^{(s)})$  = activation function based on the weighted sum

By using these equations, the non-linear activation function is computed as:

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

**f. Backward Computation**

To minimize the error in the training network, we perform the backward computation in the back propagation algorithm which helps to minimize the error between expected value and the actual output. This method is defined as follows:

1. Initiate with the output layer of the network and move back to the input layer to compute the local gradients. The computation method is shown in Eq. 4 to Eq. 6.
2. Computation of weight by using Eq. 7.
3. Update all the weights by using Eq. 8.

$$\epsilon_k^{(s)} = \begin{cases} t_k - O_k^{(s)} & s = M \\ \sum_{j=1}^{N_s+1} w_{kj}^{s+1} \sigma_j^{(s+1)} & s = 1, \dots, M - 1 \end{cases} \quad (4)$$

Where

$\epsilon_k^{(s)}$  = error value for neuron

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

$$\sigma_j^{(s)} = \epsilon_k^{(s)} f'(H_k^{(s)}), s = 1, \dots, M \quad (5)$$

where  $f'(H_k^{(s)})$  is the derivative of activation function of the network, which can be computed as partial derivation of the active function.

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

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

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

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

**II. IMPLEMENTATION**

The neuronal model has been coded in VHDL and executed. VHDL language is a blend of VHSIC (Very High Speed Integrated Circuit) and HDL (Hardware Description Language). The best approach to adapt to the issue is utilizing a top-down strategy, that is, to isolate a perplexing outline in less demanding plans or modules, every module is re-imagined with more prominent points of interest or partitioned in more subsystems. We have utilized a framework based creation as part of a simulated neuron with just three dendrites as inputs; in a brief timeframe, expected to energize a neuron are around 50. A standout amongst the most vital qualities of the neural systems is the capacity of learning. For neurons, learning is the adjustment of the actuated conduct for the cooperation. In simulated neuronal systems, the information is spoken to in the weights of the associations between the neurons. The procedure of learning suggests a few quantities of changes in these associations. The neuronal system learns adjusting the estimations of the weights of the system. The utilization of weights is imperative weights and are related to the neurotransmitters and can increment or diminish the signs that touch base to a neural connection. As mentioned before, we have characterized 3 variable weights as inputs, each weight is connected with a dendrite to enhance the capacity of learning. To start with our framework will have 3 inputs eluded to the dendrites from other neurons; 1 yield alluded to the axon. At the point when the yield would be terminated, the yield will need to create a pulse to engender it to other neuron through the axon.

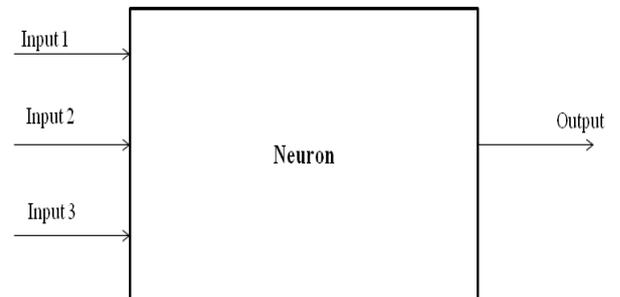


Figure 3. Input Neuron Model

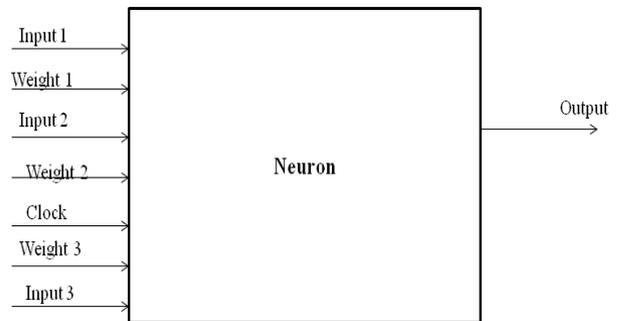


Figure 4 Input Neuron Model With Weight and Clock Inputs

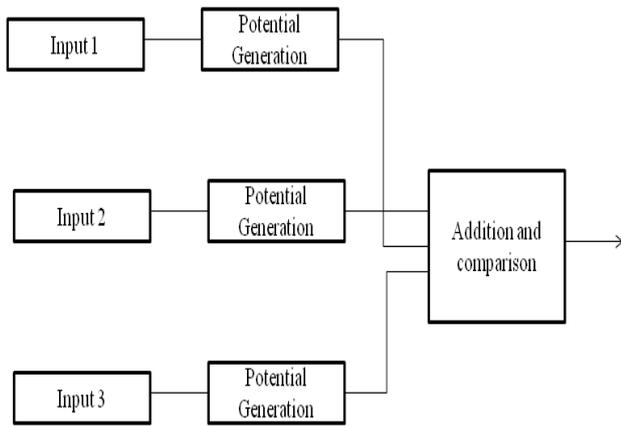


Figure 5 Input Neuron Model for Potential Generation and Input Comparison

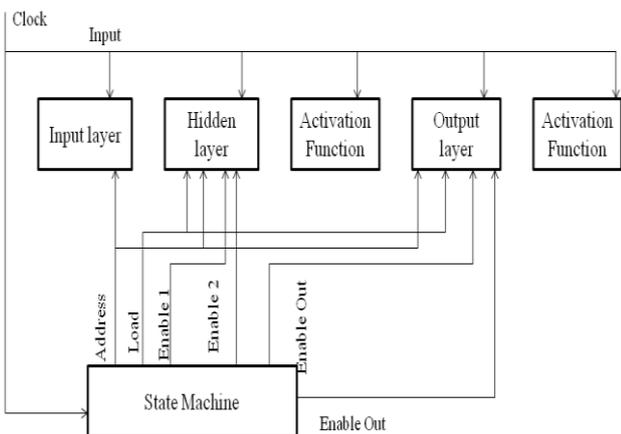


Figure 6. Network Block Diagram

In a more profound examination, we need to incorporate other essential blocks. It is important to present a sign of clock in the framework. Every one of the signs will be computerized signals; for that, it is prescribed a sign that synchronizes the worldwide framework. 3 inputs alluded to the weights of the association between different neurons and our neuron. Each weight is in connection with every dendrite. A clock sign is essential to make the framework synchronous. When the inputs are provided to the neuron model, it performs two functions (a) generation of the action according to the input and (b) comparison with time threshold. It is defined as follows Generation of the potential inputs and performance comparison is given in Fig. 5.

### III. RESULTS AND DISCUSSIONS

In this section, we discuss the results achieved for the proposed model. This model is implemented using Xilinx Virtex-II Pro kit. Table 1 presents the utilization of resources of neurons, state machines, activation functions. Similarly, Table 2 presents the resource utilization of the whole design.

Table 1. Resource Utilization

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

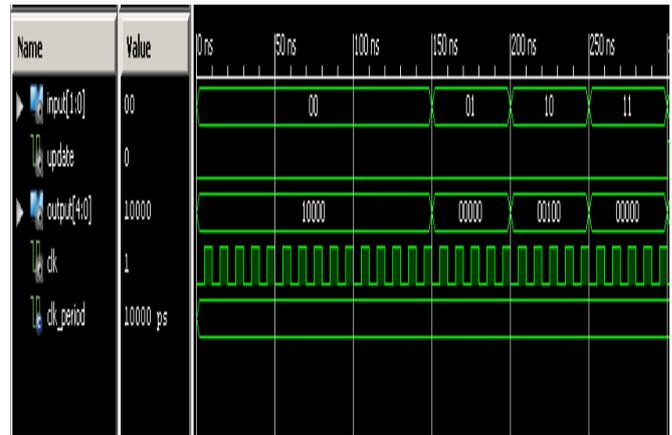


Figure 7. Initial State of the Neural Network Controller

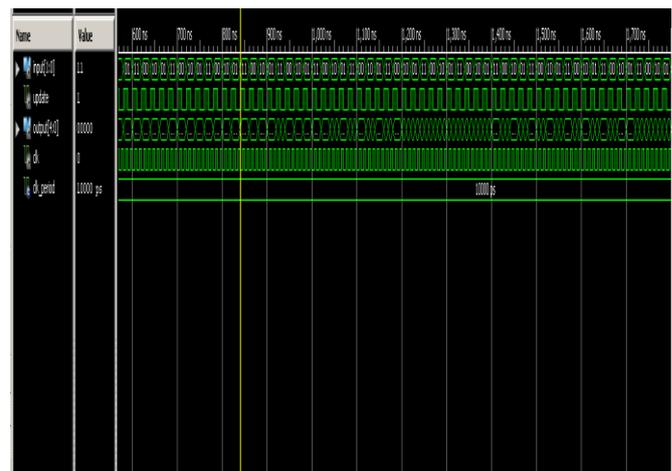


Figure 8. Training and Update State of the Neural Network Controller

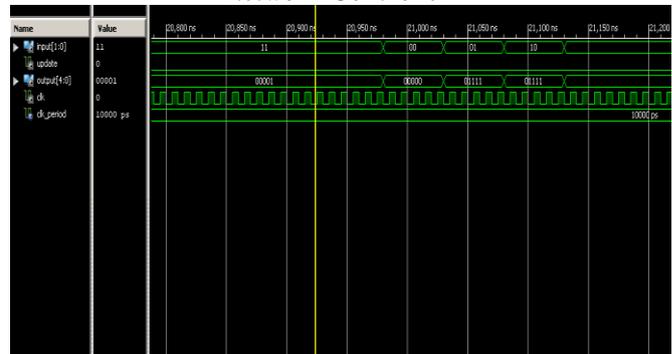


Figure 9. Output state of the Neural Network Controller

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	

**Table 3. Power Consumption Results of the Proposed Scheme**

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

**IV. CONCLUSION**

This work has presented an independent hardware design, an equipment execution of the NN algorithm for the autonomous robot controller. A pipelined hardware design approach has been used to realize the system. The design coded in VHDL was executed on a Xilinx Virtex-II Pro FPGA and performs well on two standard benchmark issues when contrasted with a typical programming usage of NN in Matlab. The framework is focused for constant applications where minimizing NN execution time is basic. The framework permits feed-forward NNs to be actualized on FPGAs bringing about high throughput. The equipment NN is created utilizing a FPGA based approach which is less demanding for specialists for their particular application. The proposed model requires less memory to store the functionality of the robot because all the functionality is stored in the register which increases the computation speed of the proposed model.

**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. L. A. Vazquez and F. Jurado, "Continuous-time decentralized wavelet neural control for a 2 DOF robot manipulator," in Proceedings of the 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE '14), pp. 1–6, Campeche, Mexico, September–October 2014
12. 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.
13. Najmuddin Aamer and S. Ramachandran, "A Novel Algorithm for Autonomous Robot Navigation System Using Neural Network" International Journal of Computational Engineering Research (IJCER), Volume, 05, Issue, 12,December – 2015.

14. Sara Bouraine, Thierry Fraichard, and Hassen Salhi. Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. Autonomous Robots, 32(3):267–283, 2012.
15. 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.
16. 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)
17. 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 , vol., no., pp.1-7, 9-12 Dec. 2014
18. 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 , vol., no., pp.2110-2115, 8-10 Aug. 2015
19. X. Jin and A. Ray , "Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments" , International Journal of Control , vol. 87 , no. 4 , pp.787 -801 , 2014
20. E. Galceran and M. Carreras , "A survey on coverage path planning for robotics" , Robotics and Autonomous Systems , vol. 61 , no. 12 , pp.1258 -1276 , 2013

**AUTHOR PROFILE**



**Najmuddin Aamer:** Completed his Bachelor’s in Electronics & Communication Engg From KBN College of Engineering, Gulbarga University, Gulbarga in 2001 and Master’s in Information and Communication Systems from Malnad College of Engg, Hassan, Visvesvarya Technological University, Belgaum in 2005. Currently pursuing PhD in Vinayaka Missions University, Salem, Tamil Nadu, India. His research areas include Mobile Robot Path Planning Using Soft Computing Techniques, Artificial Intelligence, Neural Networks and Reconfigurable Architecture’s, FPGA, CPLD.



**Dr S. Ramachandran:** Currently working as a Professor in ECE Dept., SJB Institute of Technology, Bangalore. Received his M.Tech from Indian Institute of Technology Kanpur) in 1979 and PhD from Indian Institute of Technology Madras in 2001. His research interests in a nutshell are VLSI / FPGA Design, Video Processing, Medical Electronics, Controls. He has over 75 papers in International Journals and Conferences. He has wide academic as well as industrial experience of over 30 years. He has also written a book on Digital VLSI Systems Design, published by Springer Verlag, Netherlands and NPTEL Video Talks at IITM available on YouTube.

