IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 1, Ver. I (Jan. - Feb .2016), PP 47-58
www.iosrjournals.org

High-Speed Neural Network Controller for Autonomous Robot Navigation using FPGA

Najmuddin Aamer[1], S. Ramachandran[2]
[1]*Vinayaka Missions University, Salem (Tamil Nadu),*
[2]*ECE Dept., SJB Institute of Technology, Bangalore*

Abstract : *This paper presents the implementation of a neural network controller for an autonomous robot with a neural network controller unit and a field programmable gate array. As an on-line learning algorithm of a neural network, the reference compensation technique has been implemented on a network controller unit on an FPGA chip. In this work, artificial neural network has been applied with 32-bit floating point to achieve the flexibility and accuracy for the navigation. The ANN supports reconfigurable numbers of perceptron per layer as well as supervised learning through back propagation. Mean squared error is used to measure the quality of learning as part of the Built-in Self-Test. Simulation is performed by using Xilinx 14.3 ISE simulator. The developed neural control hardware has been tested for balancing the autonomous robot while controlling a desired trajectory of a robot as a nonlinear model.*
Keywords: *Autonomous Robot, FPGA, Neural Network, Controller, High-Speed, Navigation.*

I. INTRODUCTION

The improvement of robots that can help people in their everyday assignments has turned into a marvelous course in a couple of years [1]. Portable controller frameworks open administration robots (PSR1 and 2) perform conveyance, watch, and floor cleaning occupations [2]. The aide robot gives show guide administrations at an exhibition hall. These robots are furnished with a couple of laser reach discoverers for limitation and impediment location. The PSR1 and PSR2 are driven by dynamic caster-wrote holonomic Omni-directional wheels. The aide robot utilizes customary two-wheel differential-sort wheels. Essentially, these robots offer basic control construction modeling. Some product segments were mostly changed by target robot framework. The control structural planning outline, route framework, and conduct choice system were proposed in [3] for every robot. The attention on the limitation issue whose arrangement will empower robots to complete trustworthy route in element indoor situations relies on calculation. Limitation as in [3] is a standout amongst the most essential issues for effective self-ruling route, and an awesome number of restriction systems have been proposed in this way. Numerous studies have tended to position following issues. As indicated by the International Federation of Robotics (IFR), "an administration robot is a robot which works semi or completely independently to perform administrations valuable to the prosperity of human and gear, barring producing operations" [4]. These gadgets are regularly unpredictable frameworks requiring the information of learning from various controls. The creators have been utilizing distinctive programming designing methods throughout the past 15 years, coordinating new standards in the administration robot improvement process as they rose. This has made it conceivable to accomplish fast advancement of utilizations and consequent upkeep. Way arranging is a major undertaking for a versatile robot by which it guides itself through nature on the premise of tangible data. The capability of computational vision for mechanical route is huge, and vision-based way arranging has been effectively contemplated in the most recent decade [5]. Work has advanced on two separate fronts: 1) vision-based route of indoor robot where the complete information of nature is accessible [6, 7], 2) vision-based route of open air robots where incomplete learning of the earth is just regularly accessible. Numerous current way arranging calculations have been intended for usage in programming. On account of element situations, fast arranging and recomputation of ways is important to maintain a strategic distance from crashes of robots with moving articles, especially when new questions enter nature abruptly or when moving items change their anticipated course. In such cases, the computational prerequisite surpasses the figuring force of present-day universally useful processors that actualize the way arranging calculation. It is alluring to create particular equipment coordinated arrangements which work fast and which offer extra focal points; for example, reconfigurability and conveyability. Programmable equipment gadgets, for example, field-programmable entryway clusters (FPGAs) these days give propelled highlights and assets to permit quick prototyping of framework on-chips[8].

The development of reconfigurable Field Programmable Gate Arrays (FPGA) has offered ascent to another stage of complete versatile robot control framework. With FPGA gadgets, it is conceivable to tailor the outline to fit the necessities of utilizations (for instance, investigation and route capacities for a robot). Universally useful PCs can give adequate execution when errands are not very complex. A solitary processor framework can't promise ongoing reaction (especially without impressive extra equipment), if nature is dynamic or semi-dynamic. This paper concentrates on the investigation of the versatile robot stage, with two driving wheels mounted on the same hub and a free front wheel. A FPGA-based automated framework can handle errands in parallel. A FPGA-based robot additionally enhances the single universally useful processor/PC based robot in the accompanying ranges:

1. Upgraded I/O channels. One can specifically delineate coherent outline to the processing components in FPGA gadgets.
2. Low power utilization contrasted with desktops/portable workstations.
3. Support for the consistent configuration of the non-Von Neumann computational models.
4. Support for simple verification of the rightness of the intelligent outline modules.

Wheeled mobile robots (WMRs) are more vitality proficient than legged or treaded robots on hard, smooth surfaces; and will conceivably be the first versatile robots to discover across the board application in industry, in light of the hard, smooth plant floors in existing modern situations. WMRs require less and less difficult parts and are in this way simpler to manufacture than legged or treaded versatile robots. Wheel control is complex than the incitation of multi-joint legs, and wheels cause insignificant surface harm in correlation with treads.

The portable robot comprises of numerous units:

1. Mechanics (frame, lodging, wheels),
2. Electromechanical parts and
3. Sensors.

Robots complete numerous different undertakings. Amid these errands the robot moves and arranges. While exploring, it utilizes signals from nature and the substance of its own memory to settle on the right choices. This type of route may be complex relying upon the given undertaking and issue. Frequently the objective can be detected; there is no hindrance between the objective and the robot. However, there are various times when this is not the situation, then the checking focuses must be detected and the course known. All together for the robot to have the capacity to do this, it must contain two primary parts:

1. Drive, movement and
2. Control, direction.

In our previous work we proposed an efficient algorithm for path planning and obstacle avoidance using neural network algorithm. The implementation and simulation was performed using MATLAB tool and results were compared with using proposed model for navigation and obstacle avoidance and without using proposed algorithm which shows the performance of the model in terms of navigation error [9-10]. In this paper, control hardware is designed and implemented for the autonomous robot. Control algorithms with neural network learning algorithms are embedded onto the FPGA chip whose core is the floating point processor. The functionality of the designed controller is tested to control an autonomous robot motion.

II. RELATED WORK

A few calculations for way arranging are accessible for usage on universally useful processors [10]. Notwithstanding, equipment coordinated methodologies are generally later. Specifically, equipment coordinated plans have focused on two methodologies for guide development. One methodology depends on perceivability charts, while the other depends on Voronoi outlines. A prevalent structure to speak to the earth in which a robot works is the perceivability chart. The diagram has among its hubs, the vertices of the objects [11]. Circular segments in the diagram unite the focuses x and y that can see each other. Development of the diagram goes before way finding for the robot. For most brief way calculations for portable robots and independent vehicles, one regularly considers the diminished perceivability chart rather than the complete perceivability diagram. Generally, perceivability diagram calculation has concentrated on consecutive calculations and programming usage. An immediate strategy to develop the complete chart in a domain with n hubs would require O(n3) units of time. Since on a broadly useful PC, all undertakings are executed in a consecutive manner, the perceivability diagram calculation is tedious and not proper for ongoing applications. Another equipment coordinated strategy for different periods of digression development, a focal part in perceivability diagram development is exhibited in [12]. In [13], an equipment coordinated calculation has been proposed for the development of the complete perceivability chart. In any case, the perceivability chart based methodologies expect the estimate of robot and obstructions by surrounding polygons. The guess requires extensive preprocessing.

The perceivability diagram and related methodologies are model-based, and they require displaying the earth before processing the chart. Also, the hindrances are approximated by polygonal shapes. Another valuable geometrical structure for way arranging is the Voronoi outline. A few outlines of exhibit sort architectures for the development of Voronoi chart on a situation picture are accessible in the writing [14]. The creators have connected it to way making arrangements for a jewel formed robot on an orchestrated picture containing straightforward deterrents. The graph is developed considering the cooperation's between components having a place with the same impediment and also those from distinctive obstructions, and along these lines, it has additional branches. Geometric information structures assume an imperative part in different applications. Among them is the Voronoi outline, whose applications incorporate robot way arranging, design characterization, and picture handling. The development of Voronoi graph is a basic issue in computational geometry, and the one that is built regularly is the persistent one taking into account presumption of a model for the items (polygon, bend, and so on.). The Voronoi area of an article comprises of all pixels which are nearest to that protest. The Voronoi chart is very unpredictable when a picture of genuine obstructions is considered, and the way anticipating such a graph is troublesome. Another extensive scale coordination calculation for development of the Euclidean separation based Voronoi chart is proposed in [14]. The calculation has straight time multifaceted nature (direct in picture size). The calculation includes straightforward calculations in light of nearby data, and is, consequently, manageable for equipment usage. The calculation considers the whole hindrance and not its components for the development of a Voronoi graph, thus it has no additional branches. Be that as it may, way arranging has not been endeavored utilizing the graph as a part of any of these works. Recently, a calculation has been exhibited in [15] for figuring the real way on a paired picture of a domain. The strategy builds the way from the begin point to the objective, however, on the Euclidean separation change (EDT) and Nearest Neighbor Transform (NNT) of the picture. A direct acknowledgment of the technique in equipment has been exhibited. In a prior work [15], equipment construction modeling for the separation change is accessible.

Keeping in mind the end goal to accomplish a complete path planning arrangement in equipment, interfacing systems must be intended to coordinate these architectures. A separation change changes over a twofold picture which comprises of closer view and foundation pixels into a picture where each forefront pixel has a quality relating to the base separation from the foundation. A closest neighbor change appoints the personality of the closest foundation pixel to every pixel of the picture. The separation change (DT) and closest neighbor change (NNT) have applications in picture preparing, machine vision and different spaces. In picture preparing, for occasion, separation changes discover applications in picture investigation. It is utilized for the shape examination of items in a picture. It is additionally used to figure the discrete skeleton, the discrete Voronoi graph and the Hausdorff separation for images.

Field Programmable Gate Array is picking up ubiquity in mechanical technology as of late and this is apparent through FPGA executions of picture handling descriptors, e.g., FPGA usage of pivot invariant o-daisy [16], FPGA based powerful components [17]; mapping and investigation calculations, e.g., Mechanical Exploration utilizing a FPGA based CORDIC [18], a VLSI Efficient and Fast execution of mapping utilizing FPGA [19]; Vision Processing such as Image based way arranging executed on FPGA [20], FPGA vision and SNRP Protocol [21]; Motion Control such as FPGA based IC outline for automated controllers [22], FPGA based movement control IC [23] and numerous more like Dynamically reconfigurable FPGA for robot control [24] and FPGA usage of Kalman Filter for mechanical applications [25]. The principle focal points of utilizing 2 FPGAs are ascribed to qualities like little size; capacity to reconfigure both logged off and on the web, low power dissemination, ease and rapid development.

Progressively reconfigurable architectures are assessed regarding their execution in [26-27]. How low power dissemination outlines are conceivable on FPGA has appeared in [28]. Focal points of FPGA in mechanical applications are depicted in [29]. Little measured robots with obliged assets are the need of the day and henceforth calculations are required, which can keep running on frameworks with less memory and size foot shaped impression. A base station PC has focal points of exactness and speed, however, is constrained by the remote extent to the robot and henceforth confines its working range. A portable PC can't be mounted on little robots and can be costly as well. A mobile phone processor, which can do such measure of handling is costly and expands the expense of a robot. Also an undeniable working framework is to be introduced to control it, which expands memory necessity. A microcontroller does not accommodate exactness and speed, though on a basic level a FPGA has focal points of all. Besides all these preparing units has Complex Instruction Set Computing (CISC) or Reduced Instruction Set Computing (RISC) structural engineering and there is a broad direction set. The operations depend on Fetch-Decode-Execute cycle and subsequently to a great extent consecutive in nature while on FPGA completely parallel architectures can be planned and executed. All the preparing units said above take after Arithmetic Logic Unit (ALU) based construction modeling for number juggling and sensible operations which include a ton of directions and thus clock cycles [30].

FPGA offers to create entryway level executions of number juggling circuits and subsequently a critical change in working velocity is inescapable. We demonstrate that with FPGA as the preparing unit, the execution time of the calculation, the size and cost of the robot can be decreased. Impediment evasion is a standout amongst the most key issues in versatile apply autonomy. In those frameworks where there are no other moving articles other than the robot itself static hindrance evasion is adequate. There are various calculations for static snag shirking. This can be accomplished by utilizing versatile movement primitives [31]. Ongoing randomized way arranging utilizing Rapidly-investigating arbitrary trees [32] is likewise a decent calculation for static hindrance evasion. Digression space RRT [33] is an expansion of RRT for robots subject to holonomic requirements. Arrangement space methodology is an old methodology for spatial arranging in static environment [34]. Another old methodology is ongoing impediment evasion utilizing counterfeit potential fields [35]. This is powerful in complex situations, a case which is regular when we talk of controllers. Vector field histogram system [36] is a quick ongoing shirking plan for static situations. A calculation is required which takes the dynamic way of articles and different robots under thought. Such a calculation is termed as Dynamic Obstacle Avoidance calculation [37]. There are again various methodologies for Dynamic Obstacle 3 Avoidance. Snag shirking in element environment utilizing Collision Cone methodology is a typical methodology of evasion [38]. Vision-based multi-individual following is a viable system in exceptionally intricate and dynamic scenes [39]. Inescapable impact state (ICS) is a state in environment where for a robot there is no achievable direction to maintain a strategic distance from crash. ICS idea is especially valuable for route in element situations in light of the fact that it considers future conduct of moving articles. A probabilistic methodology plan of ICS idea is appeared in [40].

Dynamic hindrance shirking for non-holonomic robots is displayed in [41]. A direction distortion plan has appeared in [42] for the situations where unanticipated snags can show up and direction must be changed on-line. Speed Obstacle (VO) methodology is one of the normal methodologies utilized for crash shirking as a part of element environment. This methodology initially presented in [43]. Speed deterrent idea has been summed up for auto like robots in [44]. Speed hindrance idea with an improved target capacity has been proposed in [44]. Utilization of speed impediment approach on a true route framework has been exhibited in [45]. A probabilistic augmentation to speed deterrent methodology for robot route among people is proposed in [46]. At the point when the quickening requirements are considered in the VO it is called Acceleration Velocity Obstacle (AVO) [47]. At the point when robots share the obligation of Avoidance among themselves just as then it is termed as Reciprocal Collision Avoidance (RCA). Corresponding impact evasion has been termed in [48], where creators propose the calculation in an n-body crash shirking case. Later same gathering of creators propose half and half corresponding speed obstructions in [49-50]. To take care of the issue of blockage because of symmetrical circumstance in thick condition a one-sided equal speed impediment approach has been proposed in [51]. Corresponding evasion for the multi-specialists route has been exhibited in [52].

A methodology which checks progression as opposed to piecewise linearity is proposed in [53]. Such a methodology contemplates movement progression imperatives alongside RCA technique. Propelled architectures like Globally Asynchronous-Locally Synchronous (GALS) architectures unite the benefit of both synchronous and offbeat architectures in one bundle. Execution change acquired in usage of a GALS construction modeling on business FPGA is displayed in [54]. Elite, adaptable and vitality proficient microchip in light of GALS chip are composed in [55]. A conceivable timing plan to address the issue of cross check synchronization in GALS building design is proposed in [56]. Use of GALS procedure in remote correspondence systems is proposed and assessed in Ref. [57]. In GALS structural planning the framework is apportioned into sub-frameworks which are exclusively synchronous, however all around these sub-frameworks associate with one another in the hand-shaking mode. The execution change of such an outline when actualized on FPGA has appeared in [58]. Outline of a GALS framework utilizing CAD devices has appeared as a part of [59].The reaction time gets 4 enhanced as any two sub-frameworks work autonomously remotely and don't need to sit tight for other sub-frameworks to finish their occupation [60]. Inner operations in the sub-framework are controlled by a clock.

III. PROPOSED MODEL FOR ROBOT NAVIGATION NEURAL NETWORK CONTROLLER

Dynamic equation for the robot is given by

$$M(r)\ddot{r} + F(r,\dot{r}) = \tau \qquad (1)$$

$M(r)$ is the matrix which contains $n \times n$ matrix of inertia of the autonomous robot, $F(r,\dot{r})$ presents the centrifugal forces, $r, \dot{r}, \ddot{r}$ shows the vector of joint angles which contains $n \times 1$ matrix, angular velocity of joints in vector form, vector of acceleration and τ represents the torque of the robot.

By keeping in mind the joint position of autonomous robot, the controller is designed as per expression:

$$C = G_p\varepsilon + G_d\varepsilon \qquad (2)$$

where $\varepsilon = q_d - q$, and gain of the $n \times n$ controller is given by G_p and G_d.

By combining Eq. (1) and Eq. (2), closed loop error model is achieved for the nonlinear loops to control the error of navigation. Owing to the computational complexity, it is difficult to implement and control as such. Thus in order to improve the controller, neural network based controller is proposed which uses the learning procedure from the feedback scheme to reduce the controller error and improve the efficiency of the autonomous robot. This scheme is depicted in Fig. 1.

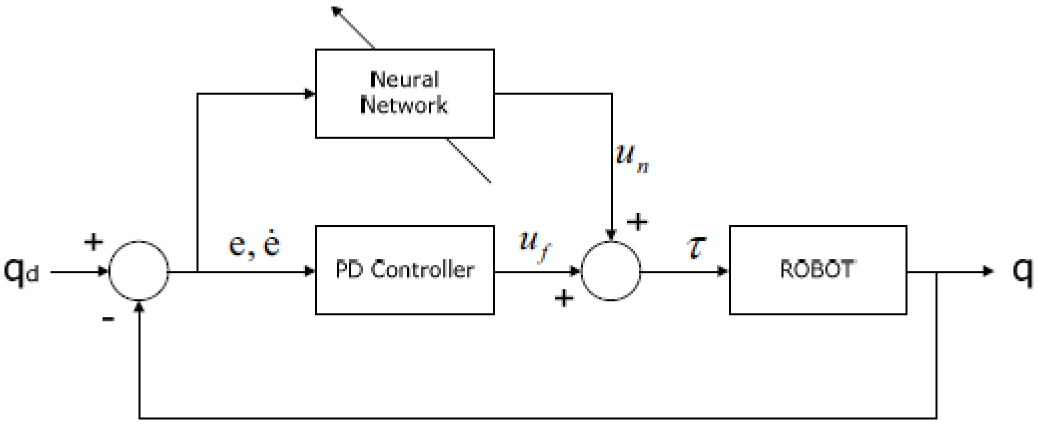

Figure-1. Block Diagram of Neural Network Robot Controller

Overall control input to the model is defined as:

$$\tau = C + C_s \qquad (3)$$

C_s is the compensating signal which is achieved by the controller.

Forward propagation of the controller

In the proposed model of neural network based controller, radial basis functionality is used which outperforms when compared to other multilayered neural network. It contains input data, hidden layer and output layers. The architecture of this is given in Fig. 2. To perform the nonlinear functionality, Gaussian function is used in the network model as per expression:

$$\mathcal{F}_j(\mathcal{I}) = \exp\left(\frac{\|\mathcal{I} - m_j\|^2}{2{c_j}^2}\right) \qquad (4)$$

where $\mathcal{I}$ denotes the input vector to the controller which is given as $\mathcal{I} = [\mathcal{I}_1, \mathcal{I}_2 \ldots \mathcal{I}_n]^T$, for all the hidden layers the mean value is denoted by m_j and similarly the covariance of the hidden layers is given by c_j.
By using these equations, the forward output layer can be computed as:

$$\mathcal{F}_o = \sum_{j=1}^{M} \mathcal{F}_j \, w_{jn} + \theta_n \quad (5)$$

$\mathcal{F}_j$ denotes the output of the hidden layer, weights between the jth and nth layer is denoted by w_{jn} and it utilizes the biasing which contains weight , denoted by θ_n.

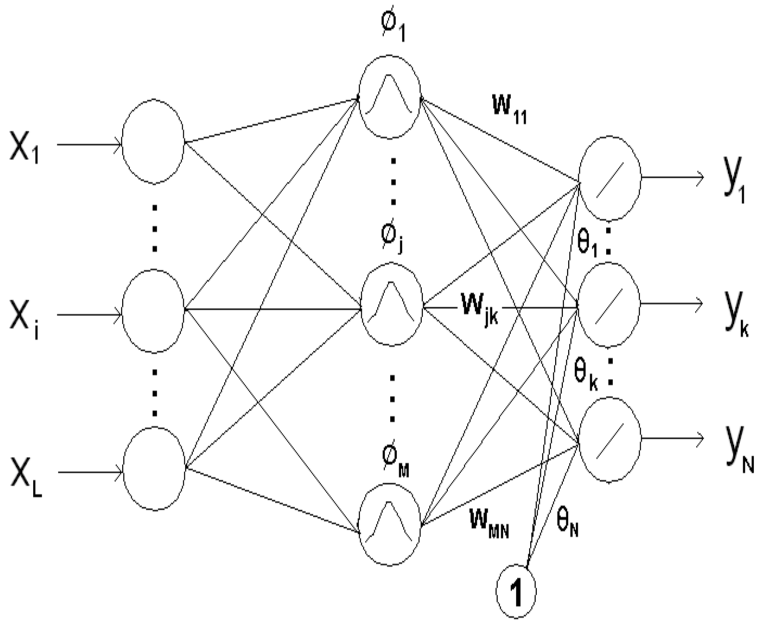

Figure 2. RBF Neural Network Structure

Backward Propagation of Neural Network Controller

The output achieved by the forward propagation is compared with the expected output to compute the errors which can be computed as:

$$\varepsilon_n = \alpha d_n - y_n \quad (6)$$

αd_n shows the desired output.
Due to the forward propagation error induced in the controller, back propagation network is utilized to reduce the errors. According to the back propagation, weights of the biases are updated and again given as feedback to the network. During this process, weights of hidden layer, output layer and function parameter units are updated based on the objective function of the network.
The objective function for the back propagation network is defined as:

$$Obj = \frac{1}{2} C^T C, C = f(r, \dot{r}) \quad (7)$$

The back propagation algorithm searches the minimum value of Eq. (7) by calculating the gradient. In the above given equation Obj shows the value of the objective function. The objective function is used to find the minimum value for achieving the best results. This best value is defined as gradient value of the network, which can be computed as:

$$g = -\zeta \frac{\delta E}{\delta w} \quad (8)$$

where δw represents the Gaussian parameters or the weights of the network and the learning rate of the network is given by ζ.

By using the above Eq., the detailed weight update can be computed as:

$$\Delta w_{jn} = \zeta \varepsilon_n \mathcal{F}_j \qquad (9)$$

where

$$\Delta \theta_n = \zeta_b \varepsilon_n$$

$$\Delta \lambda_j = \zeta_\lambda \mathcal{F}_j \sum_{i=1}^{N_n} \frac{(a_i - \lambda_j)}{{m_j}^2} \sum_{n=1}^{N} \varepsilon_n w_{jn}$$

$$\Delta m_j = \zeta_m \sum_{i=1}^{N_n} \frac{(a_i - \lambda_j)}{{m_j}^3} \sum_{n=1}^{N} \varepsilon_n w_{jn}$$

Learning rate of the model is denoted by $\zeta_w, \zeta_b, \zeta_\lambda$ and ζ_m. Then the weights updation can be defined as:

$$w(n+1) w(n) + \Delta w \qquad (10)$$

Compensation of reference

In this section, we discuss about the compensation of the error induced due to the external environment by using the neural network. By using the neural network, the input level is compensated and modified to reduce the output errors by using the same objective function.
The angle error is given as

$$\varepsilon_\theta = \theta_d - \theta \qquad (11)$$

where θ_d is the expected angle and θ is the actual angle.
Angle control is defined as:

$$ang = k_{p\theta} \varepsilon_\theta(n) + k_{i\theta} \int \varepsilon_\theta(n) dt + k_{d\theta} \varepsilon_\theta(n) + k_p \varphi_1 + k_p \varphi_2 + k_p \varphi_3 \qquad (12)$$

where φ_1, φ_2 and φ_3 give the neural network outputs.
Similarly, the position error is performed as:

$$\varepsilon = a_d - a \qquad (13)$$

where a_d is the expected position and, the actual position is given by a.

The ANN is intended to be perform such that the quantity of perceptrons per layer is effortlessly reconfigurable to meet the necessity of a particular application. Since individual weights are doled out to every association, a size of (N_I + 1) * N_H + (N_H + 1) * N_O cluster is required for the capacity of all association weights, where N_I is characterized as the quantity of perceptrons in the data layer, N_H is characterized as the quantity of perceptrons in the shrouded layer, and N_O is characterized as the quantity of perceptrons in the yield layer. As the quantity of perceptrons in a system develops, the quantity of weight stockpiling required turns out to be exceedingly expansive. Because of the restricted asset on the FPGA itself, this ANN is intended to offload such weight of memory space to an outer SRAM where storage room is generally less expensive.
It may be noted that in this ANN plan, each perceptron additionally gets a predisposition association. The predisposition association is similar to whatever other association with the exception of its quality is not a yield of a perceptron from a past layer, yet set to the consistent 1. The weight for this inclination association is likewise overhauled in the back spread procedure. Furthermore, it makes the neural system more powerful. The ANN is likewise intended to acknowledge an outside coasting guide ALU toward permit greatest adaptability. It is prescribed to utilize equipment manufactured in skimming point ALU if one is available.
A 16-bit direct input shift register is additionally required for the ANN to work appropriately. This LFSR gives vital randomized starting qualities to every one of the associations between perceptrons. Tests demonstrated that a randomized beginning system incredibly diminish the preparation length required to achieve the same level of learning. The ANN naturally creates weights between - 1.0 to 1.0 utilizing the 16-bit irregular numbers.
The procedure of the experiment is shown in Fig. 3. The FPGA takes care to calculate algorithms and to generate PWM signals. The desired trajectories for each joint are stored in the memory.

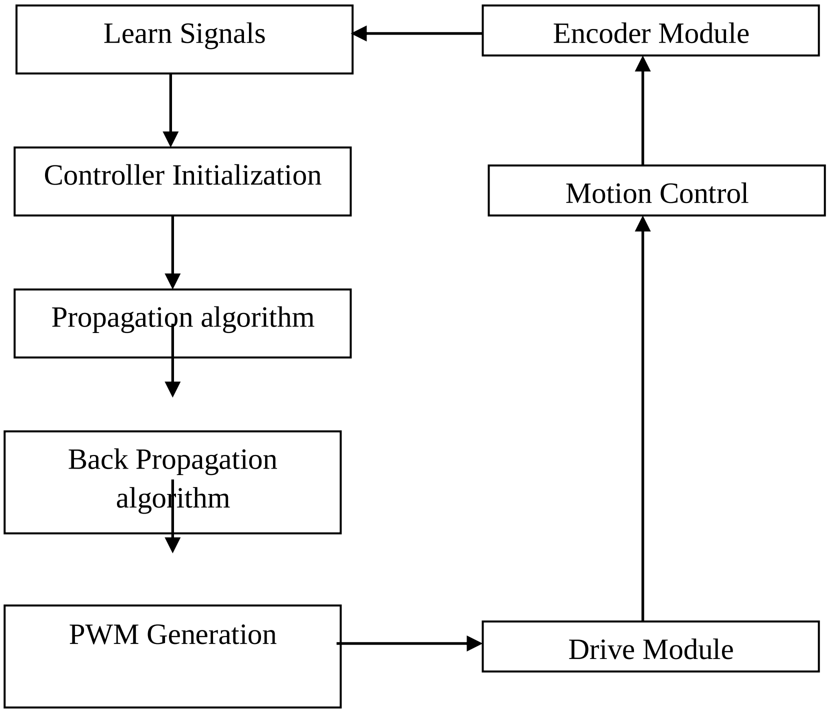

Figure 3. Motion Controller Flow for the Autonomous Robot

The following is a square graph of the ANN outline part:

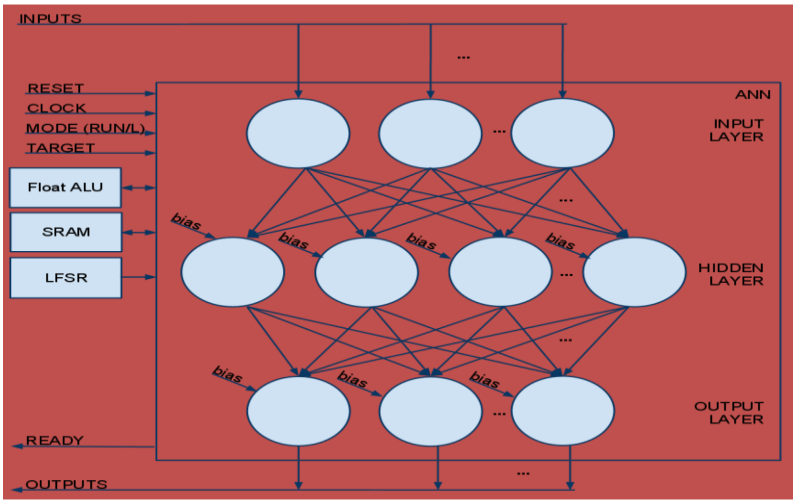

Figure 4. Artificial Neural Network Outline graph

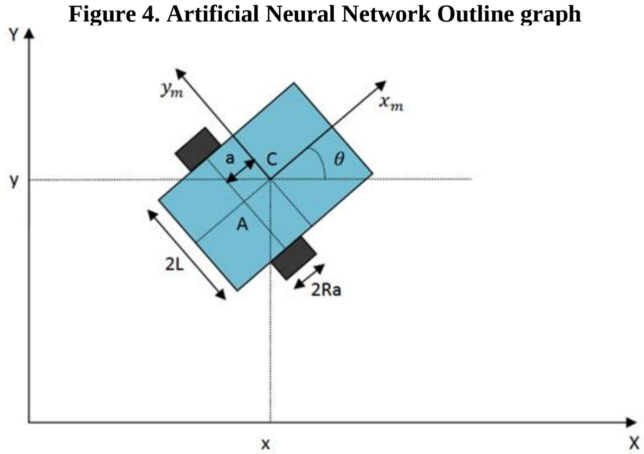

Figure 5: The Differential Drive Mobile Robot Model

The following notations will be used in this thesis:
A: axis of symmetry and wheel intersection
C: Center of mass
a: center of mass and wheel distance in x-direction
L: The distance between each driving wheel and the robot axis of symmetry in *y*-direction
R_a: Radius of wheel
$\dot{\varphi}_R$: Rotational velocity of wheel (Right wheel)
$\dot{\varphi}_L$: Rotational velocity of wheel (Left wheel)
v: The translational velocity of the platform in the local frame
ω: The rotational velocity of the platform in the local and global frames

At the point when mode is set to RUN, ANN makes the move from unmoving state to run state, where it executes numerous different states consecutively to do the accompanying:

1. For each perceptron in the shrouded layer, ascertain the weighted entirety and the yield from the sigmoid capacity.
2. For each perceptron in the yield layer, fig. 5 the weighted entirety and the yield from the sigmoid capacity.

Subsequently, ANN comes back to ideal state. At the point when mode is set to LEARN, ANN makes the move from unmoving state to run state yet with a learning banner set to HIGH. It executes numerous different states successively to do the accompanying (the initial two stages are indistinguishable to the RUN mode steps):

1. For each perceptron in the shrouded layer, compute the weighted total and the yield from the sigmoid capacity.
2. For each perceptron in the yield layer, figure out the weighted total and the yield from the sigmoid capacity.
3. Calculate mistake for each perceptron in the yield layer by subtracting yield of such perceptron from the preparation target.
4. Accumulate the mistake from yield layer to figure out the Mean Squared Error (MSE).
5. Calculate delta as the subsidiary of the sigmoid capacity duplicated by the blunder for each perceptron in the yield layer.
6. Calculate delta weight for each of the information associations for each of the perceptron in the yield layer and redesign the weight taking into account the delta weight for these associations.
7. Calculate error for each perceptron in the shrouded layer by utilizing the deltas ascertained for the yield layer's perceptron.
8. Calculate delta as the subordinate of the sigmoid capacity increased by the mistake for each perceptron in the hidden layer.
9. Calculate delta weight for each of the information associations for each of the perceptron in the concealed layer and overhaul the weight in the light of the delta weight for these associations.

A short time later, the ANN comes back to idle state.

IV. RESULTS AND DISCUSSIONS

In this section, we discuss the results of the proposed method for autonomous robot navigation using FPGA. This autonomous robot design is coded in VHDL and simulated on Xilinx Design platform. The aim of this is to achieve good performance results in terms of power, frequency and chip resource utilization. This method is applied for autonomous robot navigation using neural network. Table 1 presents the synthesis results of proposed scheme.

Table 1 Synthesis Results of the Proposed Autonomous Robot Navigation

	Used	**Available**
Number of Slices	492	63168
Number of Slice Flip-Flops	371	126336
Number of 4-Input LUTs	942	126336
Number of bonded IOBs	44	768
Frequency	358 MHz	

Fig. presents the waveform of learning process of neural network.

Learning

For the proposed model of neural network learning offline method of learning is utilized. According to this method, structures of the neural network and defined weights are fixed which can be used for particular problem. In the proposed model inputs are given to the neural network which uses activation function and weights to compute the output.

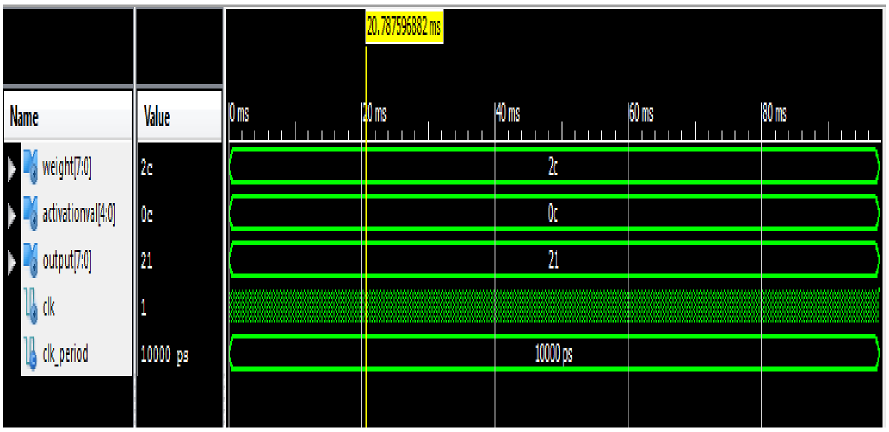

Figure 5. Learning Process of the Proposed Neural Network

For the neural network, there are some specific processes to perform the controlling, which are given as Learning, weight update, Hidden node weight update, and the weight update of output node.

Fig. 6 shows the waveform of weight update process of neural network.

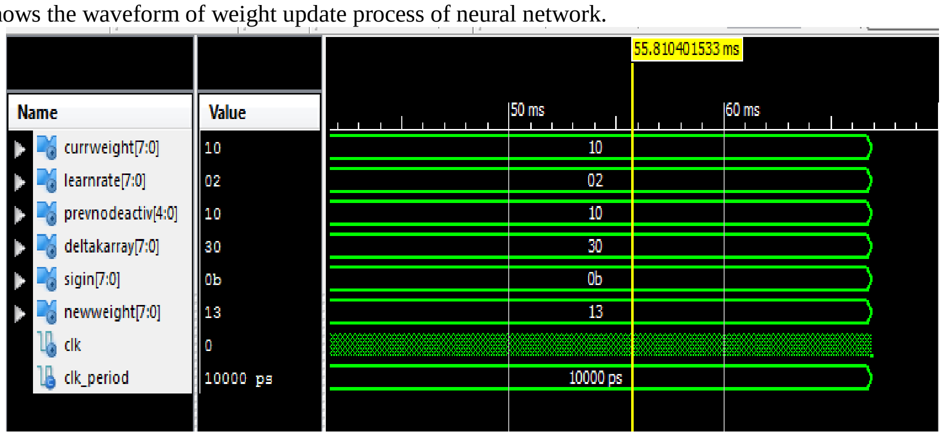

Figure 6. Weight Process of the Proposed Neural Network

Weight computation process is defined in fig.6. Initially the inputs to the neural network i.e. hidden layer, fixed weights, epochs etc. are assigned based on this the weight is updated accordingly. As it can be seen from the waveform current weight, input signal and learning rate are the main inputs and the computed weight is given by new weight.

Fig. 7 shows the waveform of hidden node weight update process of neural network.

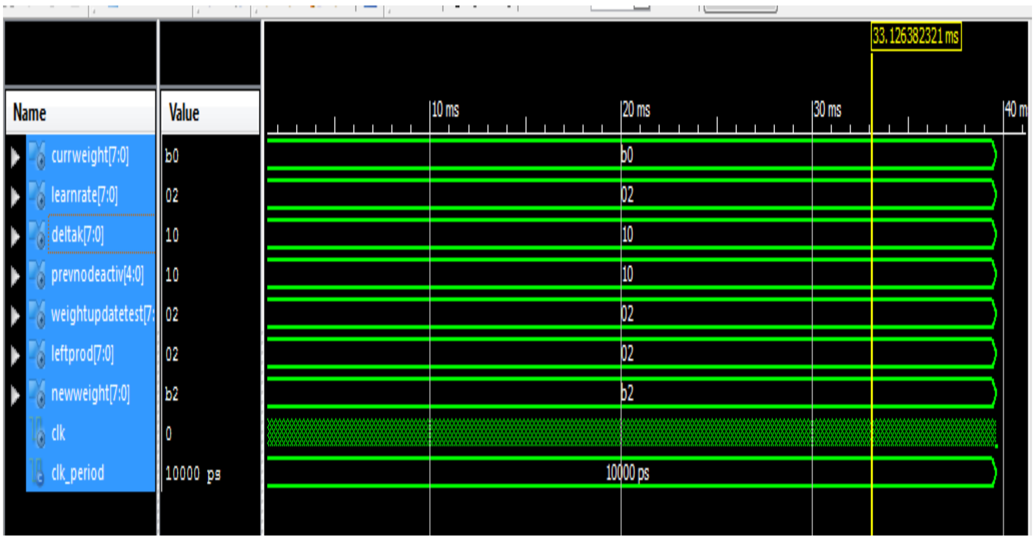

Figure 7. Hidden Node Weight Update Process of the Proposed Neural Network

V. CONCLUSION

The radial basis function network controller has been designed and embedded into the FPGA chip. The performance of the designed neural controller chip has been tested to control autonomous robot. The functionality of the designed neural network controller has been confirmed by demonstrating superior performances for various experiments. The ANN supports reconfigurable numbers of perceptron per layer as well as supervised learning through back propagation. Mean squared error is used to measure the quality of learning as part of the Built-in Self-Test (BIST). Simulation is performed by using Xilinx 14.3 ISE simulator. The developed neural control hardware has been tested for balancing the autonomous robot while controlling a desired trajectory of a robot as a nonlinear model. Simulation results show the high frequency results achieved by using this proposed model for controlling the autonomous robot.

REFERENCES

[1] Severinghaus, R. Tummala, M. and McEachen, J., "Networks for Maintaining System-Level Availability for an Exploring Robot," in Systems Journal, IEEE , vol.9, no.1, pp.98-106, March 2015.
[2] W. T. Miller, R. S. Sutton, and P. J. Werbos, Neural Networks for Control, The MIT Press, 1991
[3] M. Cristea, J. Khor, and M. McCormick, "FPGA fuzzy logic controller for variable speed generators," Proc. of the IEEE International Conference on Control Applications, pp. 301- 304, 2001
[4] Jihong Liu, Qing Guo and Xin Cheng, "Study of navigation methods based on embedded dual-encoder," in Control and Decision Conference (CCDC), 2012 24th Chinese, pp.3463-3466, 23-25 May 2012
[5] Chen Weidong, Fan Changhong and Xi Yugeng, "On-line safe path planning in unknown environments," in Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on , vol.3, no., pp.4191-4196 vol.3, 14-19 Sept. 2003
[6] P. Vadakkepat, O.C. Miin, X. Peng and T.H. Lee, "Fuzzy behavior-based control of mobile robots", IEEE Transactions on Fuzzy Systems 12 (4) (2004), pp. 559–564.
[7] S. Jung and T. C. Hsia, "Neural network inverse control techniques for PD controlled robot manipulator," Robotica, vol. 19, No. 3, pp. 305- 314, 2000
[8] Petrick, D., Espinosa, D., Ripley, R.Crum G. and Geist, A.; Flatley, T., "Adapting the reconfigurable spacecube processing system for multiple mission applications," IEEE Conference in Aerospace 2014, pp. 1-20, 1-8 March 2014
[9] 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.
[10] 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, Vol.3 Issue 12, pp. 66-79, December – 2015.
[11] Jihong Liu, Qing Guo and Xin Cheng, "Study of navigation methods based on embedded dual-encoder," in Control and Decision Conference (CCDC), 2012 24th Chinese, pp.3463-3466, 23-25 May 2012
[12] S. Jung and T. C. Hsia, "Neural network inverse control techniques for PD controlled robot manipulator," Robotica, vol. 19, No. 3, pp. 305- 314, 2000.
[13] M. E. Magana and F. Holzapfel, "Fuzzy-logic control of an inverted pendulum with vision feedback," IEEE Trans. on Education, vol. 41, No. 2, pp. 165-170, 1998.
[14] T. H. Hung, M. F. Yeh, and H. C. Lu, "A pi-like fuzzy controller implementation for the inverted pendulum system," Proc. of IEEE Conference on Intelligent Processing Systems, pp. 218-222, 1997.
[15] S. Omatu, T. Fujinaka, and M. Yoshioka, "Neuro-pid control for inverted single and double pendulums," Proc. of IEEE Conf. on Systems, Man, and Cybernetics, pp. 8-11, 2000
[16] S. Jung and H. T. Cho, "Decentralized neural network reference compensation technique for PD controlled two degrees-of-freedom inverted pendulum," International Journal of Control, Automations, and System, vol. 2, No.1, pp. 92- 99, 2004
[17] S. Jung and T. C. Hsia, "Neural network inverse control techniques for PD controlled robot manipulator," Robotica, vol. 19, No. 3, pp. 305- 314, 2000.
[18] S.-S. Kim and S. Jung, "Hardware implementation of real time neural network controller with a DSP and an FPGA for nonlinear systems," Proc. of IEEE Conf. on Robotics and Automations, pp. 4639-4644, 2004
[19] S. Jung and T. C. Hsia, "Neural network inverse control techniques for PD controlled robot manipulator," Robotica, vol. 19, No. 3, pp. 305- 314, 2000.
[20] H. Miyamoto, K. Kawato, T. Setoyama, and R. Suzuki, "Feedback error learning neural network for trajectory control for of a robotic manipulator," Neural Networks, vol. 1, pp. 251- 265, 1988
[21] S. Jung and H. T. Cho, "Decentralized neural network reference compensation technique for PD controlled two degrees-of-freedom inverted pendulum," International Journal of Control, Automations, and System, vol. 2, No.1, pp. 92- 99, 2004.
[22] S. Omatu, T. Fujinaka, and M. Yoshioka, "Neuro-pid control for inverted single and double pendulums," Proc. of IEEE Conf. on Systems, Man, and Cybernetics, pp. 8-11, 2000.
[23] S.-S. Kim and S. Jung, "Hardware implementation of real time neural network controller with a DSP and an FPGA for nonlinear systems," Proc. of IEEE Conf. on Robotics and Automations, pp. 4639-4644, 2004.

[24] M. Krips, T. Lammert, and A. Kummert, “FPGA implementation of a neural network for a real time hand tracking system,” Proc. of the First IEEE International Workshop on Electronic Design, Test and Applications, pp. 313-317, 2002.

[25] M. Cristea, J. Khor, and M. McCormick, “FPGA fuzzy logic controller for variable speed generators,” Proc. of the IEEE International Conference on Control Applications, pp. 301- 304, 2001.

[26] A. Kongmunvattana and P. Chongstivatana, “A FPGA-based behavioral control system for a mobile robot,” Proc. of the IEEE Asia-Pacific Conference on Circuits and Systems, pp. 759- 762, November 1998.

[27] J. L. McClelland and D. E. Rumelhart, “Explorations in parallel distributed processing,” The MIT Press, 1991.

[28] Bruce, J. and Veloso, M. "Real-time randomized path planning for robot navigation," Intelligent Robots and Systems, 2002. IEEE/ RS International Conference on , vol.3, pp. 2383- 2388 vol.3, 2002.

[29] Chansu Suh, Um, T.T., Beobkyoon Kim, Hakjong Noh, Munsang Kim and Park, F.C., "Tangent space RRT: A randomized planning algorithm on constraint manifolds," IEEE International Conference on Robotics and Automation (ICRA), 2011, pp.4968,4973, 9-13 May 2011.

[30] Lozano-Perez, T. "Spatial Planning: A Configuration Space Approach," Computers, IEEE Transactions on , vol.C-32, no.2, pp.108-120, Feb. 1983

[31] Khatib, O.;, "Real-time obstacle avoidance for manipulators and mobile robots," Robotics and Automation. Proceedings. 1985 IEEE International Conference on , vol.2, pp. 500- 505, Mar 1985

[32] Borenstein, J. and Koren, Y., "The vector field histogram-fast obstacle avoidance for mobile robots," Robotics and Automation, IEEE Transactions on , vol.7, no.3, pp.278,288, Jun 1991

[33] C. R. Karr, M. A. Craft, and J. E. Cisneros, “Dynamic obstacle avoidance”, in Proc. Conf. Distrib. Interact. Simul. Syst. Simul. Train. Aerosp. Envir., Int. Soc. Opt. Eng., Orlando, USA, 1995,pp. 195–219.

[34] Chakravarthy, A. and Ghose, D., "Obstacle avoidance in a dynamic environment: a collision cone approach," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.28, no.5, pp.562,574, Sep 1998

[35] Ess, A., Leibe, B.,Schindler, K. and Van Gool, L., "Moving obstacle detection in highly dynamic scenes," IEEE International Conference Robotics and Automation, 2009. ICRA '09. on pp.56,63, 12-17 May 2009

[36] Bautin, A., Martinez-Gomez, L. and Fraichard, T., "Inevitable Collision States: A probabilistic perspective," IEEE International Conference on Robotics and Automation (ICRA), 2010, pp.4022,4027, 3-7 May 2010

[37] Yi Guo and Tang Tang, "Optimal trajectory generation for nonholonomic robots in dynamic environments," IEEE International Conference on Robotics and Automation, 2008. ICRA 2008, pp.2552-2557, 19-23 May 2008

[38] Delsart, V. and Fraichard, T., "Navigating dynamic environments using trajectory deformation," IEEE/RSJ International Conference Intelligent Robots and Systems, 2008. IROS 2008. pp.226,233, 22-26 Sept. 2008

[39] P. Fiorini and Z. Shiller,”Motion planning in dynamic environments using velocity obstacles", Int. J. Robot. Res., vol. 17, no. 7, pp.760 -772 1998

[40] Wilkie, D., van den Berg, J. and Manocha, D., "Generalized velocity obstacles," International Conference on Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ, pp.5573,5578, 10-15 Oct. 2009

[41] Xunyu Zhong; Xiafu Peng; Jiehua Zhou, "Dynamic collision avoidance of mobile robot based on velocity obstacles," International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), 2011, pp.2410,2413, 16-18 Dec. 2011

[42] Kuwata, Y., Wolf, Michael T., Zarzhitsky, D., Huntsberger and Terrance L., " International Conference on Safe maritime navigation with COLREGS using velocity obstacles," Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ, pp.4728,4734, 25-30 Sept. 2011

[43] Kluge, B., "Recursive agent modeling with probabilistic velocity obstacles for mobile robot navigation among humans," Proceedings. International Conference on Intelligent Robots and Systems, 2003. (IROS 2003), vol.1, pp.376,380 27-31 Oct. 2003

[44] J. van den Berg, J. Snape, S. Guy and D. Manocha. “Reciprocal collision avoidance with acceleration-velocity obstacles”. IEEE Int. Conf. On Robotics and Automation, 2011.

[45] van den Berg, J., Guy, S. J., Lin, M. C. and Manocha, D.: “Reciprocal n-body Collision Avoidance”, Proc. Int. Symp. Robot. Res., (2009)

[46] Sadat, S.A.,Vaughan and R.T., "BRaVO: Biased Reciprocal Velocity Obstacles Break Symmetry in Dense Robot Populations," Ninth Conference on Computer and Robot Vision (CRV), 2012, pp.441,447, 28-30 May 2012

[47] van den Berg, J., Ming Lin and Manocha, D., "Reciprocal Velocity Obstacles for real-time multi-agent navigation," IEEE International Conference on Robotics and Automation, 2008. ICRA 2008, pp.1928,1935, 19-23 May 2008

[48] Rufli, M., Alonso-Mora, and J. Siegwart, R., "Reciprocal Collision Avoidance With Motion Continuity Constraints," IEEE Transactions on Robotics, No.99, pp.1-14

[49] M. Najibi, K. Saleh, M. Naderi, H. Pedram and M. Sedighi, "Prototyping globally asynchronous locally synchronous circuits on commercial synchronous FPGAs". The 16th IEEE International Workshop on Rapid System Prototyping, 2005. (RSP 2005). 63- 69, 8-10 June 2005.

[50] Zhiyi Yu, Baas and B.M., "High Performance, Energy Efficiency, and Scalability With GALS Chip Multiprocessors," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.17, no.1, pp.66,79, Jan. 2009

[51] Xin Fan, Krstic, M., Grass, E.,Sanders, B. and Heer, C., "Exploring pausible clocking based GALS design for 40-nm system integration," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012 , pp.1118,1121, 12-16 March 2012

[52] Krstic, M., Grass, E. and Stahl, C., "Request-driven GALS technique for wireless communication system," Proceedings. 11th IEEE International Symposium on Asynchronous Circuits and Systems, 2005. pp.76,85, 14-16 March 2005

[53] Xin Jia and Vemuri, R., "The GAPLA: a globally asynchronous locally synchronous FPGA architecture," 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2005. FCCM 2005, pp.291,292, 18-20 April 2005

[54] Xin Jia and Vemuri, R., "CAD tools for a globally asynchronous locally synchronous FPGA architecture," 19th International Conference on VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design, pp.6 pp.,, 3-7 Jan. 2006

[55] Xin Wang, Ahonen, T. and Nurmi, J., "Prototyping a Globally Asynchronous Locally Synchronous Network-On-Chip on a Conventional FPGA Device Using Synchronous Design Tools," International Conference on Field Programmable Logic and Applications, 2006. FPL '06, pp.1,6, 28-30 Aug. 2006

[56] Hemani, A., Meincke, T., Kumar, S., Postula, A., Olsson, T., Nilsson, P., Oberg, J., Ellervee, P. and Lundqvist, D., "Lowering power consumption in clock by using globally asynchronous locally synchronous design style," Proceedings. 36th Conference on Design Automation, 1999, pp.873,878, 1999

[57] Shyh-Jye Jou and I-Yao Chuang, "Low-power globally asynchronous locally synchronous design using self-timed circuit technology," Proceedings of 1997 IEEE International Symposium on Circuits and Systems, 1997. ISCAS '97 , vol.3, pp.1808,1811 vol.3, 9-12 Jun 1997

[58] Eun-Gu Jung; Jeong-Gun Lee; Dong-Soo Har; Kyoung-Sun Jhang, "Differential value encoding for two-phase 1-of-N delay insensitive handshake protocol in GALS systems," The 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS, vol.2, pp.II-253,II-256 vol.2, 25-28 July 2004

[59] Snape, J., van den Berg, J., Guy, S.J. and Manocha, D., "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009., pp.5917,5922, 10-15 Oct. 2009

[60] Snape, J., van den Berg, J., Guy, S.J. and Manocha, D., "The Hybrid Reciprocal Velocity Obstacle," IEEE Transactions on Robotics, vol.27, no.4, pp.696,706, Aug. 2011