INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

An FPGA based cyclic navigation algorithm for mobile robots in the indoor environment

Kishore Vennela
kishore.vennela@gmail.com
Padmasri Dr. B. V. Raju Institute of Technology, Narsapur, Telangana

M. K. Ravali
kokilaravali@gmail.com
Padmasri Dr. B. V. Raju Institute of Technology, Narsapur, Telangana

G. Manisha
gmanisha955@gmail.com
Padmasri Dr. B. V. Raju Institute of Technology, Narsapur, Telangana

T. Hema Rani
thamtamhemarani@gmail.com
Padmasri Dr. B. V. Raju Institute of Technology, Narsapur, Telangana

B. Sai Charan Rathod
charanrathod19@gmail.com
Padmasri Dr. B. V. Raju Institute of Technology, Narsapur, Telangana

ABSTRACT

Guiding a mobile robot for within the time in an indoor environment has shown the way for the study and sophistication of navigation methods. In this paper, we developed robotic structure in the indoor and developed navigational algorithm that would be helpful to move from one point to other point. This paper concentrates on the design of mobile robot control mechanism that uses an ultrasonic array for sensing random maze environment like hospitals and actuate the motor modules accordingly to complete the service. As a part of the development, we simulated in Xilinx ISE 10.4 environment using Verilog HDL programming and also synthesized on FPGA part Spartan 3AN evaluation board by considering a random environment.

Keywords: *Maze Environments, FPGA, Mobile service robots, Indoor Navigation, Guiding Protocols.*

1. INTRODUCTION

For any mobile robot, the ability to navigate in its environment either indoor or outdoor is important to complete the task allotted. In the process avoiding dangerous situations such as collisions and unsafe conditions (temperature, radiation, exposure to weather, etc.) comes first, but if it is intended to specific places in the environment, it must find those places and forward accordingly. This movement is termed to be robot navigation in which mobile robot has the ability to determine its own position in its frame of reference and plan a path accordingly towards some goal location. In order to navigate in its environment, the robot or any other mobility device requires representation, i.e. a map of the environment and the ability to interpret that representation. We may not have given much thought on how do we manage to walk across a crowded parking lot but programming a robot to do the same thing is challenging.

Ant colony algorithm [1] is based on bionics, which replicates the ant colony behavior. Ants accomplish the task of reaching from the nest to food source by exchanging information and hook up with one other in the journey. Basing on this shortest journey evaluation from the nest to food source applied in multiple fields to enhance the efficiency. The navigation of the path followed by ant group is identified in nature by foot prints by depositing pheromone on the ground. The probability that the path is hypothesis true can be adopted in practical application by estimating visiting probability of the same path by all the other ants.

Optimization of such algorithms can be implemented using neural networks [5][7] and by making use of feedback mechanism. Not only these but also potential fields can be used for obstacle avoidance along with the navigation [2]. In the implementation, every object is represented as a repulsive potential generator to move over the obstacles in the environment. The issues with such

© 2018, www.IJARIIT.com All Rights Reserved

artificial potential fields solved by simulated annealing approach, where in each step, a new solution defined for the robot and next position is chosen randomly from a set of neighbors of the current solution.

On the other hand Particle swarm optimization (PSO), another stochastic method based on the population of particles, where the robots change their own positions within search the locality [3]. The position of the robot is changed with reference to their own experience and those of its neighbors. In the iteration of the current position, every mobile robot memorizes the best position generally represented as a personal best position to compare with the current position in the environment.

In the case of an unknown environment where dynamic navigation is very much needed for particle movement in the locality, Dynamic Planning Navigation Algorithm optimized with GA(DPNA-GA) is applicable to implement obstacle avoidance as well as the navigation [4]. The pre-condition to apply this is the obstacles in the environment are fixed and the environment is unknown. Fuzzy based trajectory tracking control method [8] takes an arbitrary number of passive trailers to navigate in the environment.

In this paper, we came up with a new method of navigation that navigates the mobile robot in the indoor environment which is partitioned into sections/zones. The robot design covers every zone once and again halts at the initial position. Section-2 to describe the proposed cyclic navigation method and its hardware implementation is explained in Section-3. The implementation results and the corresponding logical transition diagram explained in Section-4 and followed by future work and references.

2. CYCLIC NAVIGATION ALGORITHM

In indoor and outdoor environments application of mobile robots has got its importance for various purposes like industry, reticulation environments etc. In this project we have come up with cyclic navigation algorithm for indoor environments explicitly that can be helpful to apply in known localities like hospitals, old age homes, ware housing, and many more.

Through the development of navigation algorithms in indoor, the involvement of unnecessary manual services can be replaced and also reliability can be emphasized. The limitations in service sectors i.e. in hospitals, where every patient has to be served timely which is not possible every time if human intervention is there, can be nullified using development of cyclic navigation for mobile robots.

The environment considered is a rectangular room. This is to simulate the normal environment of a robot, which is generally indoors. The room consists of a number of zones. Between adjacent zones, there is the possibility of a wall. When a room is created a zone is picked at random, that zone is assigned to be the goal zone. Since the room is created with random wall placements, it is possible for parts of the room to be sealed off by walls. The use of zones and walls can approximate any shape in a two-dimensional environment. A closer approximation can be achieved by making the zones sufficiently small. In the view of application, we assumed a random environment composed of partitions termed as zones and exit and entry so that every partition will have a minimum connectivity. As the total environment is divided into zones, the algorithm is developed such that it covers every zone arranged with beds to serve and this service should happen only once at each and every patient on the bed. The environment which we considered is 2-Dimensional with source and destination points both same and we made a maze environment where we assumed all obstacles as straight lines and beds as a location to serve.

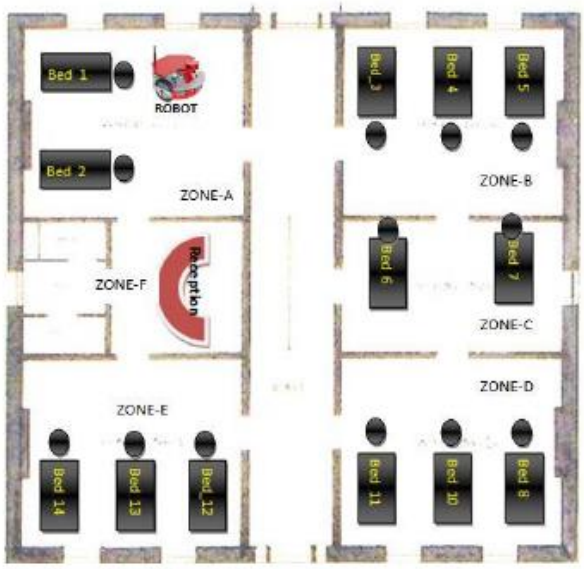

Fig. 1: Environment Assumed for Simulation

The maze environment that we considered is shown in Fig.1. The environment is partitioned with total six zones from A to F. The robot is placed in zone A which is source point and from there the robot needs to cover each and every bed present in all zones in a cyclic manner. After completion of service at every bed in every zone, it has to come to the initial point that is in zone A in the environment. In order to cover all zones by fulfilling the requirements, there are different ways as the robot moves from one zone to other within the gaps provided in each zone. The possibilities that are available for covering all the zones are noted by considering an initial point from any one of the zones we are making different possible paths from that particular point in order to cover all the zones by equally providing service to the locations to be served.

© 2018, www.IJARIIT.com All Rights Reserved

Phases in the execution of cyclic algorithm:

- Movement from ZoneA to Corridor Middle point
- Movement from Corridor Top to Corridor Middle point
- Movement from Corridor Middle point to ZoneB
- Movement from ZoneB to Corridor Middle point
- Movement from Corridor Middle point to CorridorBottom to covering every Zone
- Movement from CorridorBottom to Corridor Middle point
- Movement from Corridor Middle points to CorridorTop/ to every room in the journey.

3. HARDWARE IMPLEMENTATION OF NAVIGATION ALGORITHM

The system mainly consists of 4 ultrasonic sensors, 1 IR sensor, and 2 DC motors. The echo from the individual sensors goes as input to the Control Unit, i.e. Algorithm. Also, IR information is sent to control unit. Based on which sensor is receiving the echo, the algorithm gives the output based on the logic written. The output is in the form of wheel rotation which is nothing but the movement of the robot in a particular direction. For example, we get an echo from the front and right ultrasonic sensors, i.e. F=1, B=0, L=0, R=1. This signifies that the robot is in the right corner of a room and a left turn is required. Respective Motor movement is initiated from FPGA either by stopping one of the wheels or changing wheel around from clockwise to anti-clock wise for left and right robot wheels**.**

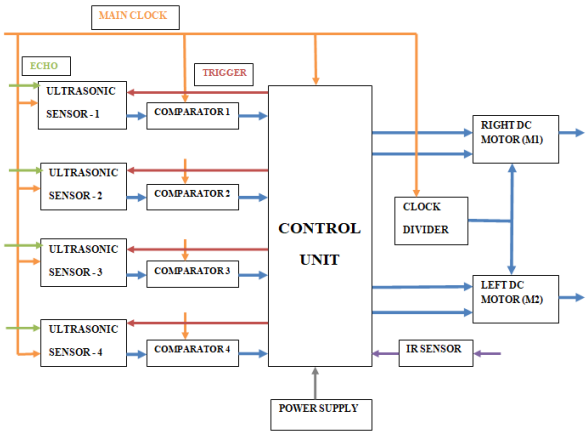

Fig. 2: Block Components of implementation

Path planning is the one that convergent on decision making to move in a particular path to the compass to the planned path. This also describes the sequence of steps to be followed during path planning. The hardware level implementation of the proposed methods of both navigation and obstacle avoidance is been done by making use of FPGAs and architecture is defined. The required state transitions are developed and are as shown in Fig.3. State transition, i.e. the robot action is based on the sensor information and current zone/locality. The default action of both robots is completely based on the navigation algorithm and the obstacle avoidance algorithm will be executed according to the array sensor information.

3.1 Robot Model

We have taken a two-layered architecture with the FPGA board on the robot chassis followed by a layer of PCB board to which 4 ultrasonic sensors have to be soldered one on each side. An IR sensor has placed in the front end of the robot, closer to the floor so that it can detect the black patches easily. The robot is equipped with two DC motors below the chassis board, one on the left and the other on the right along with two wheels and a free wheel in the front. We have used LM293D motor driver that has been connected to the motors which are powered using batteries placed on the top. These motors are mapped and controlled using the FPGA board at every stage of the robot position in the environment which is analyzed based on the four sensors logical data.

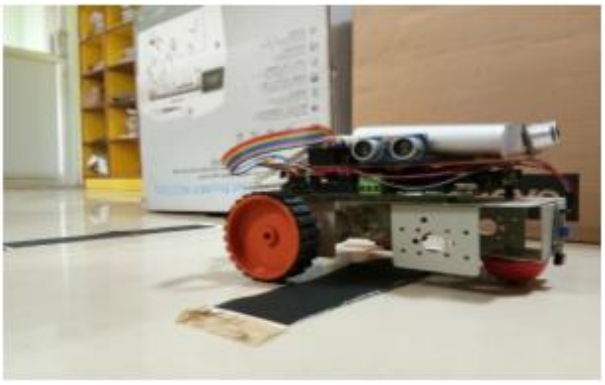

Fig. 3: Robot model

© 2018, www.IJARIIT.com All Rights Reserved

The DC motors are operated basing on supplying logical one to the positive terminal of the one end and logical zero to the negative terminal. The wheel around conditions to move forward, left , right or too backward is depicted in the Table-I.

Table I: Wheel Around Conditions Parameters

PRight	NRight	PLeft	NLeft	Action
0	1	1	0	Forward
0	1	0	0	Left
0	0	1	0	Right
0	0	0	0	Stop
1	0	0	1	Backward

3.2 Sensor Based State Transition Control

This section describes the states considered for movement of the robot from one zone to other based on the sensor data. By logical mapping of the sensor data and respective motor module switching robot covers every zone in the environment. As mentioned in Cyclic Algorithm defined in the previous section every zone is considered as one state and when the robot moves close to the wall of any portion it takes wheel around of 90 or 180 degrees according to the current position.

Initially, the robot is in the ideal state. Whenever the start signal becomes one, it moves to the state ToA. In the ToA state, it moves along the dimension of the room moving along the boundaries until it finds an exit, which is indicated by R=0 and F=0. Once it reaches this point, it takes a right 90-degree turn. When C90Degreg == Count90Deg, it moves on to state Corridor middle. At this stage, the robot starts moving forward towards the middle of the corridor. Once it reaches the middle i.e. when CCorMidreg==CountTopmid, it moves on to state CorridorTop. The robot takes a left turn from the mid-point and moves forward to the top end of the corridor. When F=1, it takes a complete 180-degree turn. When C180Turnreg==Count180Degree, it will go to state toTopForAgentA.

In state toTopForAgentA, the robot moves along the dimen-sion of the room A to check if there is any opening. It keeps moving forward and whenever it receives L=0, it shifts to state InroomA. Else if it is L=1 even after completing the entire di-mension of the room, i.e. when CRoomAreg==CountRoomA, it moves to state toMiddleForAgentB. In case it enters room A, it checks for data from IR sensor and it stops for a few seconds when it detects a black patch. Once the tine is over, it takes a 180-degree turn and comes out of the room and takes a 90 degree left turn and continues the journey along the corridor. This procedure is the same for all the rooms.

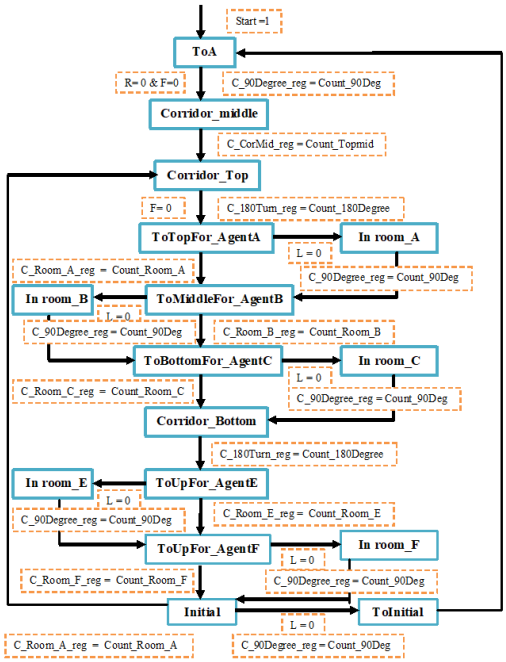

Fig. 4: State Transition Diagram for implementation

In state toMiddleForAgentB and toBottomForAgentC, the same method is followed as in-state toTopForAgentA. After toBottomForAgentC state, the robot reaches the bottom of the corridor. It is now in CorridorBottom state, where F=1 and it requires to take a 180-degree turn. When C180Turnreg == Count180Deg, it moves to toUpForAgentE followed by toUpForAgentF. The procedure is same as for every other room. Once the robot has completed the dimension of the room F as well, it moves to Initial state.

In the initial state, it will be in the upper part of the corridor again. If L=0, then it recognizes an opening and it takes a left 90-degree turn and enters ToInitial state after which it gets back to ToA state. Hence, it is given the name cyclic navigation because it reaches its starting point again. In case of L=1, while in state Initial, it traverses along the dimension of the room to reach the CorridorTop state. This is how our cyclic navigation algorithm works.

© 2018, www.IJARIIT.com All Rights Reserved

The control algorithms developed such that process the sensor information so as not only to navigate the robot but also to avoid the obstacle. The motor modules controlled through the obstacle avoidance and navigational methods defined according to the data from the sensor array. In our implementation pair of DC motors is controlled by specifying positive and negative terminal supplies activation from the control unit synchronously according to the required direction of movement.

4. RESULTS

Navigation of the mobile robot designed using FPGA considers the walls of the individual partition of the environment as obstacle is simulated and synthesized using Cyclic Navigation Algorithm. This implementation is simulated using Xilinx ISE 2010.4 and is synthesized and net list is mapped to Spartan 3AN Evaluation Board.

4.1 Simulation

As in Fig. 5, the every state specified with unique one hot coding and robot movement from one zone to other zone is represented as one state to other state transition. Every transition is decided by the logical value of sensors indicated by F, B, L, R for forwarding, Backward, Left and Right sensors respectively. The sensor with being activated on supply and triggered by a pulse of 10usec by which sound generation will take place. Upon detection of an obstacle, the echo is sensed and processed using FPGA by specifying the constant or range of distance as a control switch.

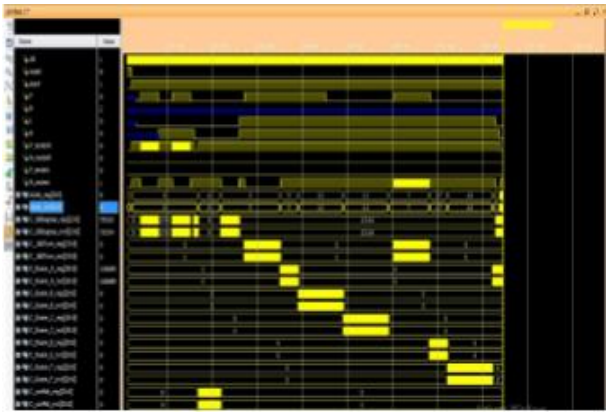

Fig. 5: Simulation of State Transition

From the simulation results, we can observe the several state transitions that occur for a robot to move from the initial position i.e. ideal state to various rooms along the corridor and getting back to its starting location. Hence, it is called cyclic navigation algorithm because the starting and the ending point is the same.

4.2 Hardware Results

The robot starts from state 8. In this state, it tries to come out of the initial room towards the corridor. It reaches state 6 i.e. it in the corridor and moves to the middle of the corridor based on the count values given. Now, it is in state 5, where the robot turns left and moves to one end of the corridor till it is at a distance of 1 meter from the wall, where it takes a complete 180-degree turn again based on the 180 degrees counter. It is now ready to check which rooms are open, in case it is open it enters the room, else it moves further to the next room along the corridor.

Fig. 6: Entry

Fig. 7: Black Label Identification

Fig. 8: Right Turn

Fig. 9: Wall Detection

© 2018, www.IJARIIT.com All Rights Reserved

Fig. 10: Towards Exit

Fig. 11: Left movement

Fig. 12: Exit

After the 180 degrees turn, the robot is in state 9, where it traverses the length of the room A minus 1 meter and checks if there is any entry based on the left sensor. If the left ultrasonic sensor gives a zero, then it means there is an opening and the robot enters room A and will be in state 11, else if it is a one, it goes to state 10. While in room A, it checks for IR data and if a black area is detected it stops for a few seconds and it then takes a 180-degree turn and comes out to the corridor. Now in state 10, it again moves along the entire dimension of room B, and checks for the opening. If not available, it goes to state 13 and moves along the dimension of room C minus 1 meter. If L=0, it enters room C else it goes to state 7, where it has reached the end of the corridor and takes a 180-degree turn, now ready to check the rooms on the other side. This process continues while in other states also and finally lands near the initial room.

5. CONCLUSION

Navigation of the mobile robot in the indoor environment implemented in this work and same conception can be transformed into service robots in hospitals, old age homes, and industrial applications. The robot agent moves according to the sensory data and respective motor module control in the indoor localities to complete the service assigned. As a next step, the algorithm can be adopted to navigate robot basing on the emergency token request from any sub-locality in the environment.

6. REFERENCES

[1] Path Planning Method for Mobile Robot Based on Ant Colony Optimization Algorithm Yuwan Cen; Choingzhi Song; Nenggang Xie; Lu Wang Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on 3-5 June 2008, Singapore

[2] Experimental Evaluation of Robot Path Planning by Artificial Potential Field Approach with Simulated Annealing M.G. Park', M.C. Lee'SICE 2002. Proceedings of the 41st SICE Annual Conference, 5-7 Aug. 2002, Osaka, Japan

[3] Particle Swarm Optimization-Based Method for Navigation of Mobile Robot in Unstructured Static and Time-Varying Environments, Irfan Jabandzi c1 and Jasmin Velagic2,2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol) Barcelona, Spain, Sept. 7-9, 2016

[4] Dynamic Planning Navigation Technique Optimized with Genetic Algorithm Atila V. F. M. de Oliveira 1 and Marcelo A. C. Fernandes2 Department of Computer Engineering and Automation DCA Federal University of Rio Grande do Norte UFRN Natal, Brazil 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robot control

[5] An Effective Vector-driven Biologically-motivated Neural Network Algorithm to Real-time Autonomous Robot Navigation Chaomin Luo1, Simon X. Yang2, Mohan Krishnan1, and Mark Paulik1 2014

[6] IEEE International Conference on Robotics and Automation (ICRA) Hong Kong Convention and Exhibition Center May 31 - June 7, 2014. Hong Kong, China

[7] Neural-network-based Fuzzy Logic Tracking Control of Mobile Robots Chaomin Luo Department of Electrical and Computer, University of Detroit Mercy, Michigan, USA 2017

[8] A Fuzzy Logic-based Reactive Navigation Algorithm for Mobile Robots X. Yang, Student Member, IEEE, M. Moallem, Member, IEEE, and R.V. Patel, Fellow, IEEE Proceedings of 2005

© 2018, www.IJARIIT.com All Rights Reserved