Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VII (2012), No. 1 (March), pp. 135-146

Adaptive Neuro-Fuzzy Controler With Genetic Training For Mobile Robot Control

O. Obe, I. Dumitrache

Olumide Obe
"Politehnica" University of Bucharest, Romania
E-mail: oluobes@gmail.com

Ioan Dumitrache
"Politehnica" University of Bucharest, Romania
E-mail: ioan.dumitrache@cncsis.ro,
idumitrache@ics.pub.ro

Abstract:
In this paper, we investigate the use of adaptive techniques in the optimization of navigation of Khepera mobile robot in an unstructured and dynamic environment. We optimize the performance of our simplified fuzzy controller using neural network that utilizes genetic algorithm learning. The adaptation of the system involves the tuning of the control rules thereby trimming the control actions, and adjusting the fuzzy controller output gain. We realised an improved performance in our adaptive neuro-fuzzy controller with genetic training for various implemented behaviours on the robot.

Keywords: Khepera, fuzzy controller, neuro-fuzzy controller, navigation, genetic algorithm.

1 Introduction

Navigation and obstacle avoidance are very important issues for the successful use of an autonomous mobile robot in a dynamic and unstructured environment. Mobile robot researchers aim to build an autonomous and intelligent robot which can plan its motion in a dynamic environment. A successful use of an autonomous mobile robot depends on its controller. Mobile robot control is difficult as they are subjected to non-holonomic (non-integrable) kinematic constraints involving the time derivates of configuration variables [12] and dynamic constraints. Both analytical like potential field method as well as graph-based techniques have been used to solve the navigation problems of robot involving both static and dynamic obstacles. But, all such methods may not be suitable for on-line implementations due to their inherent computational complexity and limitations. Mobile robot researchers have carried out various researches in this direction using various intelligent techniques methods such as fuzzy logic, neural network and genetic algorithm and their different hybrids. Because of the non-linear kinematics of the robot, the uncertainty in sensors readings, and unstructured environmental constrains in the control of mobile robot navigation; researchers have found fuzzy logic as one of the best intelligent technique for handling the constraints. However, fuzzy logic needs tuning for optimal performance. Hand tuning is very difficult and time consuming therefore there is need for automation of the tuning process. The process of tuning requires learning brought about by training or adaptation of the robot to adapt to its dynamic environment. The poor learning capability of fuzzy logic is compensated for by hybridizing fuzzy logic with other soft computing techniques with excellent learning features such as neural network. In this paper, we present an adaptive neuro-fuzzy controller with genetic algorithm learning for the navigation of Khepera mobile robot.

2 State-of-the-art of Control of Mobile Robots

A mobile robot is a situated and embodied agent endowed with mobility; and operates autonomously, communicating with, and exploring its environment. Mobile robots are complex systems functioning in real world environments thus, making it uneasy to design an adaptive control system that can control robots to act as desired. Task based decomposition of robot control by Brooks' [8] subsumption architecture that makes easy design of robots to realize multiple behaviours, respond to multiple sensors and incrementally extension has been successfully applied in mobile robot [9].Researchers in robotics use theories and concepts from intelligent control theory described as behaviour-based control as an alternative to conventional robot control since real world cannot be accurately modeled. Because of the need to operate in unknown environments, mobile robots demand much higher level of intelligence in order to learn to adapt successfully to its ever changing environment. Intelligence in a mobile robot is considered as an adaptive behaviour that makes a robot adapt to and acts intelligently in its environment.

Intelligent control is an act of directing a complex system to a goal [1].The intelligence is the property of a system that emerges when procedures of focusing attention, combinatorial search and generalization are applied to the input information so as to receive the output results [2].Consequently, Murphy [3]defines an intelligent mobile robot as a situated agent with a mechanical structure that operates autonomously.

Intelligent systems strategies like fuzzy logic, neural network, and genetic algorithms have been used to endow robot with intelligent capability to navigate its environment autonomously. Various hybrids of these strategies have also been implemented successfully on mobile robots. In this work, we employ hybrid geno-neuro-fuzzy intelligent strategies to control Khepera mobile robot.

3 Fuzzy Control of Mobile Robot

Fuzzy control is one of the intelligent control techniques that pertain to the realization of intelligent control systems. Fuzzy control provides a mechanism for incorporating human-like reasoning capabilities and computation in control systems. The linguistic variables are used to mimic the human action into a system more closely than traditional control. Fuzzy logic is a logical system that aims at a formalization of approximate reasoning [10].These can be represented as the concept of a linguistic variable, canonical from, fuzzy if-then rule, fuzzy quantifiers, and modes of reasoning.

3.1 Fuzzy Controller Design

In this paper, we design a simple fuzzy logic controller for Khepera mobile robot for obstacle avoidance and wall following behaviours.

Our Fuzzy logic controller architecture is as shown in figure 1:

Where, r = sensors input from the robot and $U_{ext} = [5000, 5000]$ i.e. the maximum robot's motor speed for stability (for our design), which is equal to $35mms^{-1}$ real speed.U_{fc} is the Fuzzy Controller output; U = the resultant speed for the robot; and Y = the robot output control signal.

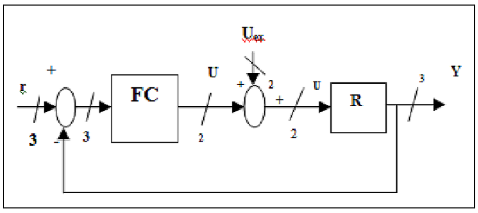


Figure 1: Architecture of our Fuzzy Logic Controller for the Mobile robot showing the number of inputs and outputs parameters at each processing stage.

Obstacle Avoidance Fuzzy Controller

We design the obstacle avoidance controller to allow the robot to avoid colliding with any object along its path to its target. The controller will empower the robot to maneuver its way around any obstacles without colliding with them. By using Mamdani fuzzy logic approach, we define three sensor input variables representing relative distance between the robot and any object in its environment. These include: Left Distance (LD), Right Distance (RD), and Front Distance (FD). Since the robot rarely uses the back sensors because, the robot usually moves forward, we exclude back sensor readings in our design. The physical domains over which these input variables are defined are determined as:

$LD = Max(S2, S3)$;
$FD = Max(S4, S5)$;
$RD = Max(S6, S7)$.

Where $Si(i = 2, 3...7)$ are the sensor values normalized within the interval [0,1]. All these input variables have the same base variable length.

Figure 2.1 shows our defined membership function distributions of the input variables. For simplicity, we use two types of shapes for our membership function. These include the z-shaped and s-shaped membership functions shape types. Two linguistics variables (or grades of distance) from the robots' sensors are considered: Not Detected (ND), and Detected (D). We use z-shaped membership function type for the ND grade and s-shaped membership function type for the D membership function. These membership function types are the variant of Gaussian membership functions that have been proved experimentally by [11] to be of better performance for the same robot navigation problems.

The output fuzzy variables are the motor speed of the robot's wheels, represented by Left Velocity, LV and Right Velocity, RV. For the output fuzzy variables, we use three linguistic variables (S-Slow, Z-Zero, and F-Fast) for each of the fuzzy output variables. We use Gaussian membership function type for each of the linguistic variables. The simplicity in the design is to reduce complexity. This is at the cost of granularity which we belief would be compensated for by the optimization of fuzzy controller using neural network with genetic training. This design invariably reduces the number of fuzzy rules as expected, to 8 rules. Some of the 8 rules are shown in Table 1.

4 Neural Network Design

A layer network is used for our Neural Network as shown in figure 2.

Where,

Xi = sensors input (the context and the last two history inputs each of three pair)

$Y = f(W * \begin{bmatrix} 1 \\ X \end{bmatrix}) + bias)$, bias=1

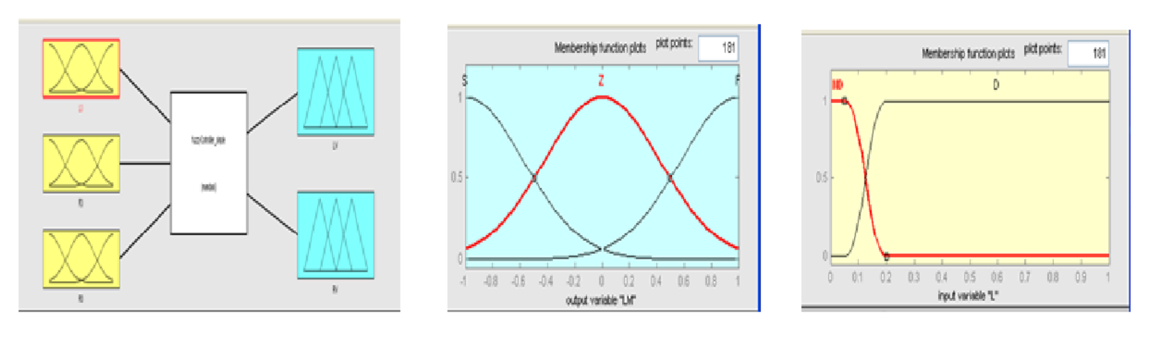


Figure 2: Fig.2.1 Fuzzy Controller design Fig.2.2: Input MFs Plot Fig.2.3: Output MFs Plot.

	INPUT SENSORS IF			OUTPUT MOTOR WHEELS THEN	
Rule No.	LD	FD	RD	LV	RV
1	ND	ND	ND	Z	Z
2	ND	ND	D	Z	F
3	D	ND	ND	F	Z
4	ND	D	–	S	F
5	D	D	–	F	S

Table 1: Fuzzy Rule Table for Khepera III Fuzzy Controller

$f(s) = \frac{1}{1+e^{-s}}$
$S = W_{01} + W_{11} * X_1 + \cdots + W_{19} * X_9$
$W \in U_{5,10}(R)$ (5 neurons for fuzzy weights with 9 sensors inputs and a bias).

We design a network of 5 neurons representing the 5 fuzzy rules weight to be optimized. The inputs are the sensor readings from the 3 fuzzy input variables (LD, FD, and RD). In other to ensure that our robot collides not with the obstacles considering the simplicity in our fuzzy design, we decided to keep history of the last two sensors readings with the context readings of the robot as the input to our NN. This allows the robot to keep history of the last two previous inputs values from the sensors thereby increasing accuracy of decision of eliciting obstacle avoidance bahaviour.

The input of our NN is the sensor values from the robot and the output is the 5 optimized rule weights to optimize fuzzy logic controller performance for obstacles and wall following behaviours.

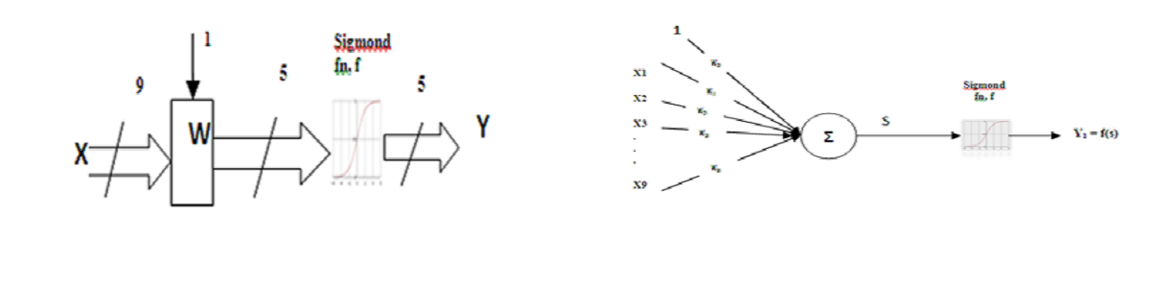


Figure 3: A layer Neural Network Architecture for our Network design

Genetic Algorithm for Training

We employ GA as the learning algorithm of Neural network weight which invariably optimize NN performance resulting in NN optimizing fuzzy Logic rules weights and thereby optimizing the Robot navigation in terms of wall following and obstacle avoidance behaviours.

Initially, 50 random weight matrix of 20 population size for each individual are generated and made to evolve for 100 generations. This was tested on the robot to get new 50 different weights. For each generation, a performance index, ϵ to evaluate the fitness of each chromosome is computed in terms of the number obstacles avoidance performed successfully. We set a fixed path to be traversed by the robot for each generation. Each chromosome is evaluated based on this performance measure. The evaluation of the chromosome fitness determines either to accept the chromosome as potential part of the new population for reproduction in the next generation. The fit chromosomes are set as part of the new population. For the next generation, 20 new populations are generated by using genetic operators like crossover and mutation. Two point crossovers were used and the mutation factor was set at 0.05. With the best 10 chromosome and 10 new chromosomes, the robot is tested. In this way the algorithm continues till the optimum obstacle performance is obtained.

Figure 4 below shows the flow chart for our genetic algorithm implementation for the optimization of NN weights.

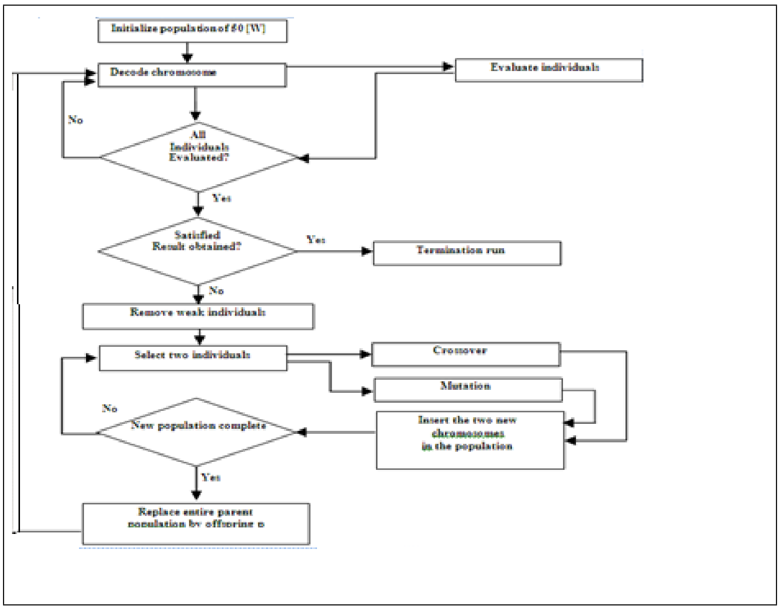

Initialize population of 50 [W]
Decode chromosome
Evaluate individuals
No
All Individuals Evaluated?
Yes
Satisfied Result obtained?
Yes
Termination run
No
Remove weak individuals
Select two individuals
Crossover
Mutation
No
New population complete
Insert the two new chromosomes in the population
Yes
Replace entire parent population by offspring s

Figure 4: Flow chart for our genetic algorithm implementation for the purpose of optimization of NN weights.

5 Neuro-Fuzzy Control of Mobile Robot with Genetic training Design

We design a Neuro-Fuzzy controller for our mobile robot control using genetic training(figure 5).

The overall design of our Neuro-Fuzzy controller with genetic algorithm learning is as illustrated in figure 9 above.

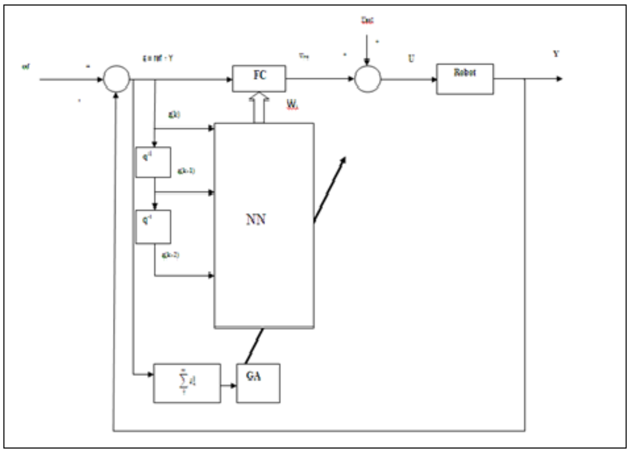


Figure 5: Geno-Neuro-Fuzzy Controller for Khepera Mobile Robot

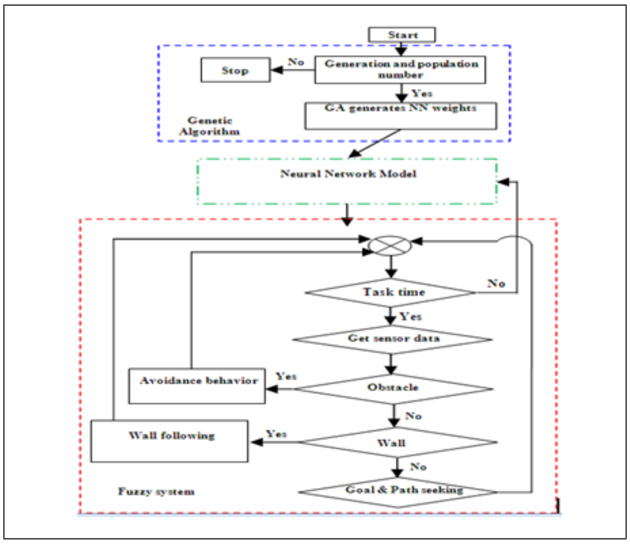


Figure 6: A schematic diagram showing flowchart of the the Khepera Mobile Robot motion planning scheme of our robot control using geno-neuro-fuzzy controller.

The overall Neuro-Fuzzy controller with genetic training has the sensors values of the robot from the fuzzy variable parameters as the reference input, all initialized to zero. The sensor readings from the robot are the inputs to the fuzzy controller and the neural network. The outputs of the fuzzy controller are the speed values for the left and right motors of the robot. The external speed input, U_{ext} represents the possible maximum set hardware speed for the Khepera robot in other to obtain stable performance. The resultant output speed, U is used for the navigation control of the robot. The difference in the robot output speed Y and input speed is computed and feed back as error, ϵ. This error serves as the fitness function to measure the fitness of each individual chromosome in the GA.

6 Result and Discussion

In the developed neuro-fuzzy approach, training is done off-line with the help of Genetic algorithm. The computer simulation is carried out by considering an arena (KiKS arena) with number of obstacles. Apart from smoothness in path traced by the robot, we have decided to use other index of performance measures to evaluate the performance of the optimization results. These include the error measure implemented in our program to monitor the difficulties in obstacles avoidance. Each experiment is run in the same environment with equal number of obstacles along the robot's path. Hence, we can measure performance in terms of difficulties in circumventing obstacles by the robot by penalizing or rewarding the actions. This measure is what we call error penalty (EP) in this work. The objective function is to minimize this error. Hence, the smaller the error, the better the performance output.

Fuzzy Controller Obstacle avoidance simulation results

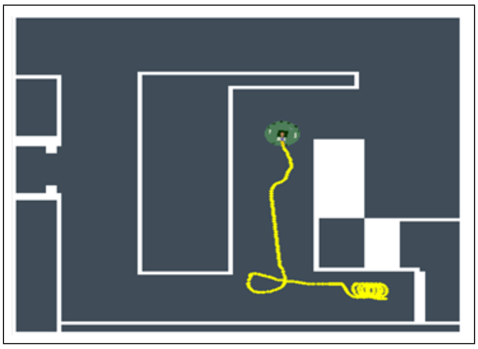


Figure 7: FL controller output (EP = 57.99)

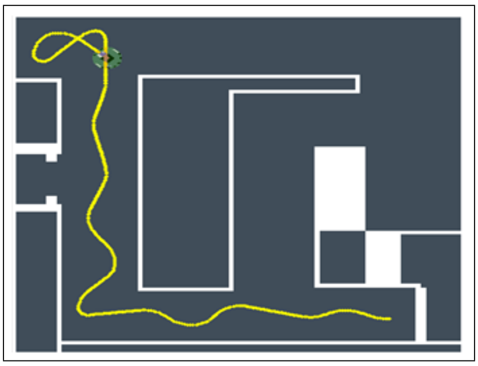


Figure 8: FL controller output after manual tuning of rule base (EP =12.58)

From figure 7, it is obvious that even though our fuzzy controller succeeds in avoiding obstacles, the robot was trapped initially and could not move forward until after several loops. In other to improve the robot's performance, we decided to manually tune our fuzzy controller by adjusting some rules consequences of the rule base. The result of the manual tuning is made obvious by the improved performance observed in figure 11 below. The robot can now move forward to the target without being caught in a loop. We equally observed reduced error measured from 57.99 to 12.58.We further improve on the fuzzy controller by manually adjusting the membership sets of the fuzzy controller. We notice improved performance in the path traced by the robot and the number of error encountered in obstacle avoidance behaviour (from 57.99 errors to 12.58 errors). This result is shown in figure 9 below.

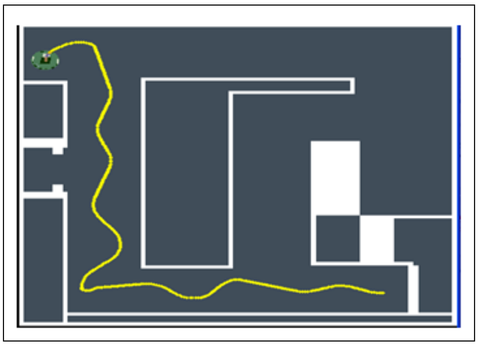


Figure 9: Fuzzy controller after hand tuning of rule base and membership function sets (EP = 12.58)

From the fuzzy controller performance results presented above, we could not reach better performance in terms smoothness of path traced by the robot by manual tuning. The lack in granularity is responsible for the zig-zag movement of the robot along its path. Therefore, we decided to employ neural network with genetic training to fine tune the navigation performance. Because of the simplicity in the fuzzy design, in terms of number of rules in the rule base, we observed that the optimal result would be better realized by deciding on what rules to be executed at an instant of time instead of further tuning of rule base and membership function parameters. Hence, we focus on the automatic tuning of the rule weights. The idea is based on the fact that we can train an NN model using a genetic training to evolve an optimal rule weights for the navigation control, wall following, and obstacle avoidance behaviours for our robot. The neuro-fuzzy controller with genetic training designed in this paper was put to test and we have realized the results below.

Neuro-fuzzy with genetic training for obstacle avoidance result

In this section, we present the implementation result of our neuro-fuzzy controller with genetic training.

The first weights set generated by our neuro-fuzzy with genetic training were used in the neuro-fuzzy controller. The simulated output result of this implementation is shown in figure 13. Apart from smoothness in path traced by the robot and error penalty (EP), we equally utilize the readings as observed by the simulation trace output report window of the KiKS simulator as shown in figure 14.

The report gives information on the number of Forward, Backward, and Straight moves of the robot. With this information, we can easily measure the performance based on number of straight moves made by the robots and also find the ratio of straight moves to forward moves (called performance ratio in this work) in other to compare the performance of each set of weights set generated in each generation. This performance ratio helps to further justify path smoothness traced by the robot.

Hence, our overall desire is to obtain performance output result with the smoothest path traced by the robot; minimum error penalty (EP); and maximum performance ratio (PR).

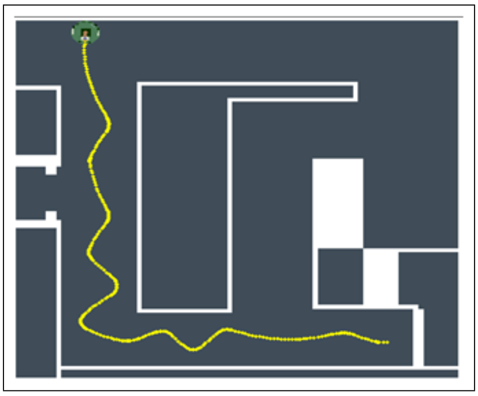


Figure 10: Geno-Neuro-Fuzzy Controller output after first evolution of rule base weights. Error penalty: 64.89; Forward: 181cm; Straight: 84.6cm; Performance ratio: 0.46

The result of implementation of the first weight set output of the neuro-fuzzy controller with genetic training on the robot navigation is shown in figure 10. By comparing figure 9 of manual tuned fuzzy controller with figure 10, there is no much difference in terms of path traced smoothness of the robot.

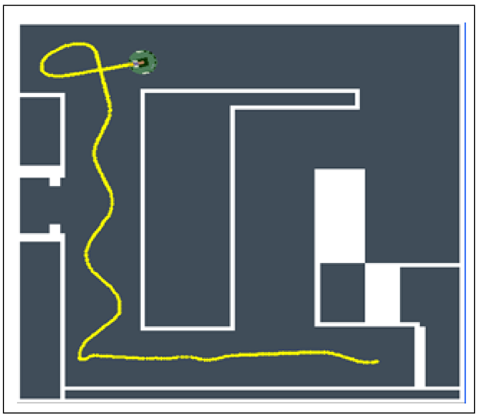


Figure 11: Geno-Neuro-Fuzzy Controller output after 50 evolutions.Error penalty: 12.79; Forward: 181.0cm;Straight: 84.6cm; Performance ratio: 0.46

After several evolutions, we tested the best rule weight output on the robot and observed the performance output in figure 14. We noticed a greater improved performance in terms of path traced by the robot and error penalty (from 64.89 to 12.79).

The performance output after further evolutions of rule weights by the genetic algorithm for the NN learning results in greater improved performance measures in terms of error penalty (from 12.79 to 9.7390), and performance ratio (0.46 to 0.66) as indicated in figure 15.

Further evolution results show a little drop in performance as shown in the figure 16. After several generations by the GA, we could not get a rule weights set of better performance than

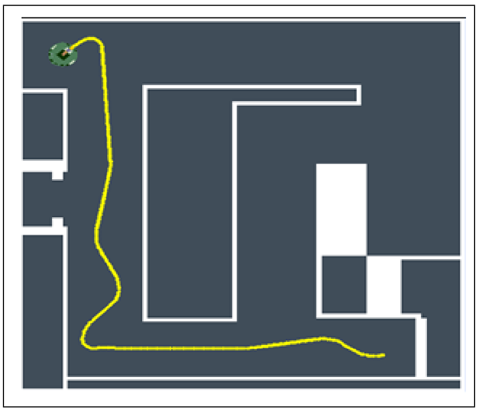


Figure 12: Geno-Neuro-Fuzzy Controller at 95th evolutions. Error penalty: 9.7390; Forwards: 146.7cm; Straight: 96.6cm; Performance Ratio = 0.66

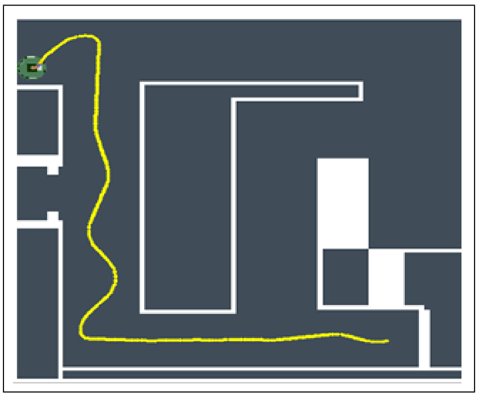
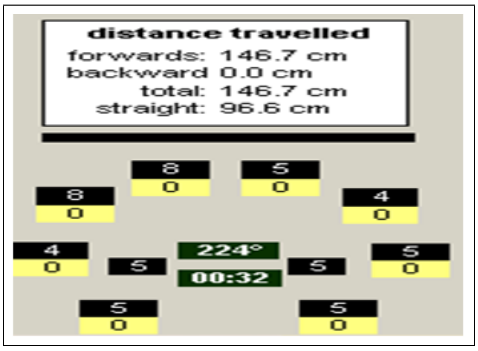


Figure 13: Geno-Neuro-Fuzzy controller output after 100 evolutions of rule weights.Error penalty: 12.20; Forward: 154.3cm;Straights: 96.8cm; Performance ratio = 0.63



distance travelled
forwards: 146.7 cm
backward 0.0 cm
total: 146.7 cm
straight: 96.6 cm

Figure 14: KiKS Window to monitor Simulation details

the one shown in figure 15 (95^{th} generation). This shows that we have reach the optimum performance for the rule weight generation for our experiment. Hence, we can use the NN model for this optimal rule weight set to obtain an optimal performance in terms of obstacle avoidance behaviour and navigation of our Khepera robot. The NN model with this optimal netweight values for rule weights is judged as the best NN model for our research result.

7 Conclusion

In this paper, a simple neuro-fuzzy strategy with genetic training has been proposed to drive a Khepera mobile robot. With this approach, we have been able to tune the rule weights of our fuzzy controller in order to guide our robot. The proposed neuro-fuzzy controller is a layer neural network along with an evolutionary (genetic) learning algorithm. This system has been successfully implemented in simulation obtaining satisfactory results which is better than fuzzy controller output. The lack of granularity in the membership functions has been to a large extent compensated for by the adaptive hybrid neuro-fuzzy with genetic training controller. This is obvious from the smoothness of path traced by the hybrid system compared to the path traced by the fuzzy controller outputs. We have equally adapted the neuro-fuzzy controller with genetic training to work on real Khepera III robot. The result was satisfactory. The simulation and real implementation on the Khepera III robot have shown the efficacy of using hybrid systems in improving mobile robot navigational performances in terms of obstacles, wall following and path tracking behaviours.

Bibliography

[1] I. Dumitrache, 1996, Intelligent Control of industrial Robots, Mediamira Press, Cluj, 1996.

[2] I. Dumitrache, 2000, Intelligent Autonomous Systems, Revue Roumaine des Sciences techiniques - Electrotechique et Energetique, Vol. 45, No.3, Pp. 439-453, Bucharest.

[3] R.R. Murphy, 2000, Introduction to AI Robotics,MIT Press.

[4] I. Dumitrache and M. Dragocea, 2006, Some problems of advanced mobile robot control,CEAI, Vol. 7, No. 4, pp. 11-30

[5] G. Campion , G. Bastin, 1996, Structural Properties and classification of Kinematics and Dynamic Model of wheeled Mobile Robots, IEEE Transaction on Robotics and Automation, Vol. 12,No. 1.

[6] H.R. Everett, 1995, Sensors for Mobile Robots,A.K. Peters, Ltd.

[7] X.Q. Chen, Y.Q. Chen, and J.G. Chase, Mobiles Robots-Past Present and Future, Intech.

[8] R. A. Brooks, 1986, A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, RA-2(1):14-23.

[9] R. A. Brooks, 1999, Cambrian Intelligence: The Early History of the New AI, The MIT Press, Cambridge, Massachusetts.

[10] L.A. Zadeh, 1994, Fuzzy logic, Neural networks, and soft computing, Communications of the ACM, 37(3):77-84.

[11] O.O.Obe and I. Dumitrache, Fuzzy Control of Autonomous Mobile Robot, U.P.B. Sci. Bull., Series C, Vol. 72,Iss. 3, 2010

[12] M.Jackson Phinni et. al, Obstacle Avoidance of a wheel mobile Robot: A Genetic-neuro-fuzzy approach, IISc Centary-International Conference on Advances in Mechanical Engineering (IC-ICAME),Bangalore, India, July, 2008