World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:1, No:10, 2007

Navigation of Multiple Mobile Robots using Rule-based-Neuro-Fuzzy Technique

Saroj Kumar Pradhan, Dayal Ramakrushna Parhi and Anup Kumar Panda

Abstract—This paper deals with motion planning of multiple mobile robots. Mobile robots working together to achieve several objectives have many advantages over single robot system. However, the planning and coordination between the mobile robots is extremely difficult. In the present investigation rule-based and rule-based-neuro-fuzzy techniques are analyzed for multiple mobile robots navigation in an unknown or partially known environment. The final aims of the robots are to reach some pre-defined goals. Based upon a reference motion, direction; distances between the robots and obstacles; and distances between the robots and targets; different types of rules are taken heuristically and refined later to find the steering angle. The control system combines a repelling influence related to the distance between robots and nearby obstacles and with an attracting influence between the robots and targets. Then a hybrid rule-based-neuro-fuzzy technique is analysed to find the steering angle of the robots. Simulation results show that the proposed rule-based-neuro-fuzzy technique can improve navigation performance in complex and unknown environments compared to this simple rule-based technique.

Keywords—Mobile robots, Navigation, Neuro-fuzzy, Obstacle avoidance, Rule-based, Target seeking

I. Introduction

THE problem of navigation in a changing, unknown environment is presented in this paper. Multiple robotic systems are becoming more and more significant in industrial, commercial and scientific application. Applications in the area of service robotics demand a high degree of system autonomy. Which robots without learning capabilities will not be able to meet? Problems such as co-ordination of multiple robots, motion planning and co-ordination of multiple robotic systems are generally approached having a central (hierarchical) controller in mind.

Control and communication methods for robotic systems have been investigated by various researchers, which are described below:

Gürman [1] has described in detail the neural network models RuleNet and its extension. RuleNet is a feed forward network model with a supervised learning algorithm, a dynamic architecture, and discrete outputs. He has achieved results in the simulation and experimental environment. Li *et al.* [2] have presented neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments. The simulation experiments show that the proposed neuro-fuzzy system can improve navigation performance in complex and unknown environments. Barfoot *et al.* [3] have discussed a newly developed Adaptive Fuzzy Behavioural Control System. That has been designed for use with an autonomous mobile robot. They have shown their results on both experimental and industrial applications in which their new control system was applied. Their results have shown that the robot can avoid simple obstacles only. Experiment results done on a single mobile robot confirms their technique.

Jelena *et al.* [4] have considered a rule-based fuzzy controller and a learning procedure based on the stochastic approximation method. They have considered the radial basis function neural network and have shown that a modified form of this network is identical with the fuzzy controller, which they claim it as a neuro-fuzzy controller. Acosta *et al.* [5] have used a neuro-fuzzy technique to steer a mobile robot. The results of the approach are satisfactory (i.e. avoiding the obstacles when the mobile robot is steered to the target). Marichal *et al.* [6] have presented a neuro-fuzzy approach in order to guide a mobile robot. They have shown that such a neuro-fuzzy system is successful in the control of a single mobile robot only.

Althoefer *et al.* [7] have reported a navigation system for robotic manipulators. The navigation method is a combination of fuzzy logic and neural networks. They successfully applied this technique to robot arms in different environments. Nefti *et al.* [8] have applied neuro-fuzzy inference system for mobile robot navigation in partially unknown environment. The experimental results confirm the meaningfulness of the elaborated methodology when dealing with navigation of a mobile robot in partially unknown environment.

Tunstel *el al.* [9] have discussed operational safety and health monitoring of critical matters for autonomous field mobile robots. De Souza *et al.* [10] have proposed a reusable

Manuscript received October 28, 2005.

Saroj Kumar Pradhan is with Department of Mechanical Engineering, N.I.T., Rourkela, Orissa, 769008, India (corresponding author.:Tel.: ++919437202842; Fax: ++916612472926;
e-*mail*:saroj_pradhan2000@rediffmail.com)

Dayal Ramakrushna Parhi is Assistant Professor of Department of Mechanical Engineering, N.I.T., Rourkela, Orissa, 769008, India (e-mail: dayalparhi@rediffmail.com)

Anup Kumar Panda is Assistant Professor Department of Electrical Engineering, N.I.T., Rourkela, Orissa, 769008, India(e-mail: akpanda@nitrkl.ac.in)

architecture for rule-based systems described through design patterns. The aim of these patterns is to constitute a design catalogue that can be used by designers to understand and create new rule-based systems. A hybrid control architecture combining behavior based reactive navigation and model based environment classification has been developed by Na *et al.* [11]. The effectiveness of the proposed architecture has been verified through both computer simulation and an actual robot called MORIS (Mobile Robot as an Intelligent System). Dietrich *et al.* [12] have discussed a general architecture for rule-based agents and described the method to realise the navigation control with the help of semantic web languages. McIntosh *et al.* [13] have described a simple 'proof of concept' rule-based system. They have developed to contribute methodologically to management-oriented modelling of vegetated landscapes. They have not specifically used rule-based technique for navigation of mobile robot.

Navigation of multiple mobile robots in presence of static and moving obstacles using rule-based technique is presented in this work. First, the extraction of a set of navigation rules is extracted from the data base through See 5 algorithm. The rules are used on their own to control the navigation of multiple mobile robots. Also the rules can be combined with another tool to produce a hybrid navigation technique for control of mobile robots. Both techniques are described here. Simulation results using 'ROBPATH' (Appendix-A) are presented to demonstrate the performance of the proposed approach.

II. ANALYSIS OF CONTROL ARCHITECTURE

A. *Rule-based technique*

The rules used for navigation of multiple mobile robots are generated by induction from examples. Approximately one thousand examples are fed to See5 (Appendix - B). See5 is a rule induction program, within the Clementine data mining software package [14].

The examples present the situations encountered by a robot while moving in a multi-robot, highly cluttered environment, and the actions that each robot has to take to avoid colliding with other robots as well as with fixed obstacles. Each example consisting four input elements specifying the distances of the obstacles to left, to right, and in front of the robot and the target angle and an output element giving the change in the steering angle of the robot being required in response to the input data. Some of the rules are mentioned below:

Attributes:

Left distance, Front distance, Right distance, Target angle & Change in steering angle

1) 60,60,10,12,-7
2) 60,50,32,12,-9
3) 60,26,54,12,-5
4) 60,23,18,14,-8
5) 60,24,42,14,-7
6) 60,23,52,14,-5
7) 60,25,28,16,-7
8) 60,56,46,16,-9

' – ' means change in steering angle towards left and ' + ' means change in steering angle towards right.

Rule 6 described that if the robot is finding an obstacle at a distance 60 to the left, 52 to the right, 23 to the front and the robot is moving at target angle of 14, then change in steering angle required to reach the target as well as to avoid the obstacles is -5. In this rule, the distances are in cm each and target angle and the change in steering angle are in degrees.

In this exercise one thousand seven hundred situations are fed to See5. From these situations, See5 yields 54 rules. Out of 54 rules nine rules are listed below:

Rule 1

If (left distance > 20 cm) and (front distance <= 12 cm) and (right distance <= 30 cm) and (target angle > 10˚) and (target angle <= 16˚)
Then Change in steering angle = -7˚

Rule 2

If (left distance > 12 cm) and (front distance <= 20cm) and (right distance <= 30 cm) and (target angle > 14˚) and (target angle <= 20˚)
Then Change in steering angle = -7˚

Rule 3

If (left distance > 18 cm) and (front distance <= 12 cm) and (right distance > 30 cm) and (right distance <= 48 cm) and (target angle > 12˚) and (target angle <= 24˚)
Then Change in steering angle = -7˚

Rule 4

If (left distance > 18 cm) and (front distance <= 12 cm) and (right distance > 30 cm) and (right distance <= 48 cm) and (target angle > 14˚) and (target angle <= 26˚)
Then Change in steering angle = -9˚

Rule 5

If (left distance <= 28 cm) and (front distance <= 30 cm) and (right distance > 30 cm) and (right distance <= 48 cm) and (target angle > 10˚) and (target angle <= 22˚)
Then Change in steering angle = -9˚

Rule 6

If (left distance > 18 cm) and (front distance <= 32 cm) and (right distance > 48 cm) and (target angle > 10˚) and (target angle <= 26˚)
Then Change in steering angle = -5˚

Rule 7

If (left distance > 25 cm) and (front distance <= 32 cm) and (right distance <= 31) and (target angle > 20˚) and (target angle <= 30˚)
Then Change in steering angle = -8˚

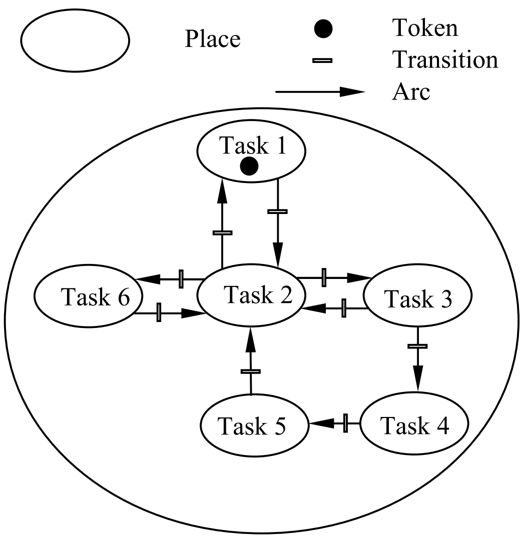
Rule 8

If (left distance > 15 cm) and (front distance <= 30 cm) and (right distance <= 30) and (target angle > 12˚) and (target angle <= 24˚)
Then Change in steering angle = -8˚

Rule 9

If (left distance > 20 cm) and (front distance <= 24 cm) and (right distance <= 29) and (target angle > -30˚) and (target angle <= -19˚)
Then Change in steering angle = -8˚

B. Petri Net modelling to avoid collision among robots



Task 1 Wait for the start signal
Task 2 Moving, avoiding obstacles and searching for targets
Task 3 Detecting conflicts,
Task 4 Negotiating,
Task 5 Checking for conflict and executing movements,
Task 6 Waiting

Fig. 1 Petri Net Model for avoiding inter-robot collision

C. A. Petri [15] first developed Petri Net model. The detail principle of the Petri Net Model to avoid robot collision is described below.

It is assumed that, initially, the robots are in a highly cluttered environment, without any prior knowledge of one another or of the targets and obstacles. This means the robot is in state "Task 1" ("Wait for the start signal"). In Fig. 1, the token is in place "Task 1".

Once the robots have received a command to start searching for the targets, they will try to locate targets while avoiding obstacles and one another. The robot is thus in state "task2" ("Moving, avoiding obstacles and searching for targets").

During navigation, if the path of a robot is obstructed by another robot, a conflict situation is raised. (State "Task 3", "Detecting Conflict"). Conflicting robots will negotiate with each other to decide which one has priority. The lower priority robot will be treated as a static obstacle and the higher priority robot as a proper mobile robot (state "Task 4", "Negotiating"). As soon as the conflict situation is resolved, the robots will look for other conflicts and if there is no other conflict they will execute their movements (state "Task 5", "Checking for conflict and executing movements").

If a robot meets two other robots already in a conflict situation, its priority will be lowest and it will be treated as a static obstacle (state "Task 6", "Waiting") until the conflict is resolved. When this is done, the robot will re-enter state "Task 2".

III. Analysis of Rule-Based Technique

In the analysis various rules are extracted and subsequently experimented separately and combinedly. The rules are extracted from the data base through the rule-based technique 'See 5' (Appendix-B). The rules extracted by See 5 are experimented individually.

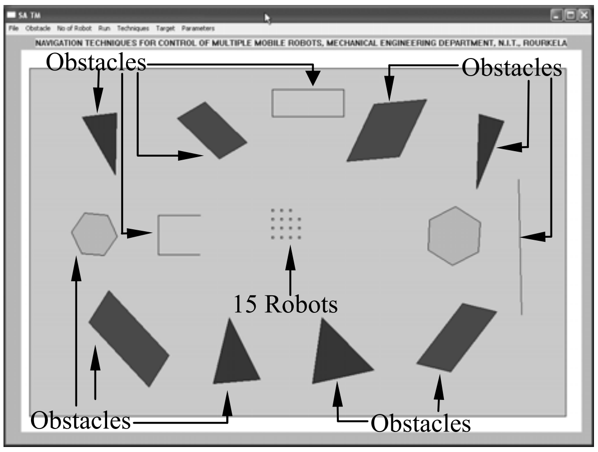

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Obstacles
Obstacles
15 Robots
Obstacles
Obstacles

Fig. 2 Scenario before applying all the rules simultaneously

The set of all the 54 induced rules performs better. From the simulation results it is observed that some robots are not able to avoid obstacles. The scenario is shown in Fig. 2 and Fig. 3.

In addition to the above rules, some rules are incorporated to fine tune the rule-based system. The new rules are required, as the induced rules do not cover all situations encountered by the robots.

The additional rules are listed below:

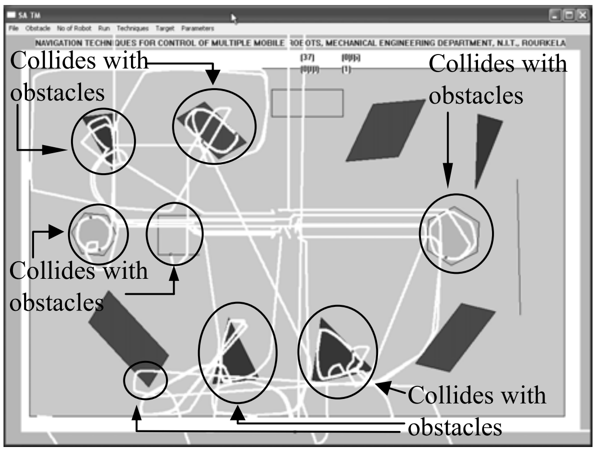


Fig. 3 Final scenario when are the rules are applied simultaneously

Rule 55

If (left distance <= 20 cm) and (front distance > 20 cm) and (right distance > 25 cm) and (right distance<=30 cm) and (target angle > 23˚) and (target angle <= 40˚)
Then Change in steering angle = 8˚

Rule 56

If (left distance > 15 cm) and (front distance <= 10 cm) and (right distance <= 30cm) and (right distance >23 cm) and (front distance > 20 cm) and (right distance > 20 cm) and (left distance>20 cm) and (target angle > 18˚) and (target angle <= 24˚)
Then Change in steering angle = 10˚

Rule 57

If (left distance <= 5 cm) and (front distance > 20 cm) and (right distance <= 30cm) and (right distance >20 cm) and (front distance > 20 cm) and (right distance > 20 cm) and (left distance>20 cm) and (target angle > 10˚) and (target angle <= 18˚)
Then Change in steering angle = 4˚

Rule 58

If (left distance <= 29 cm) and (front distance <= 20 cm) and (right distance <= 30cm) and (right distance > 18 cm) and (front distance > 20 cm) and (right distance > 20 cm) and (left distance>20 cm) and (target angle > 23˚) and (target angle <= 34˚)
Then Change in steering angle = 14˚

Rule 59

If (front distance <= 20 cm) and (right distance <= 40cm) and (right distance > 30 cm) and (front distance > 20 cm) and (right distance > 20 cm) and (left distance> 20 cm) and (target angle > 10˚) and (target angle <= 23˚)
Then Change in steering angle = 6˚

Rule 60

If (front distance < 20 cm) or (right distance < 20 cm) or (left distance < 20 cm)
Then Change in steering angle = 7˚

Rule 61

If (front distance < 14 cm) or (right distance < 14 cm) or (left distance < 14 cm)
Then Change in steering angle = 13˚

After including the above 7 rules the robots are able to navigate and reach their target efficiently in a highly cluttered environment.

IV. HYBRID TECHNIQUE

The set of rulers forms the core of a pure rule-based controller. This set of rules combined with other tools to yield a hybrid controller. Here rule-based technique has been integrated with neuro-fuzzy technique to improve the navigation of robots in highly cluttered environment.

The resulting architecture is shown in Fig. 4. The role of the rule-based controller is to estimate the initial steering angle for the neuro-fuzzy controller. Then the initial steering angle is fed to neural controller along with the distances of obstacles to the left, to the right and in front of the robot. The neural network (Appendix-C) used here is a back propagation multilayer perceptron having four layers. The input layer has four neurons. Out of the four neurons three are meant to receive the distances of the obstacles in front and to the left and right, where as fourth neuron is for the initial steering angle. The output layer has a single neuron meant to produce the second estimate of steering angle. The output of the neural network i.e., the second estimate steering angle is fed to the fuzzy controller (Appendix-D) along with the information concerning the distances of the obstacles in front and to the left and right of the robot. The output of the fuzzy controller is to compute the velocities of the driving wheels. From the velocities of the driving wheel the steering angle is calculated to avoid obstacles and reach the target. Gaussian Membership function is used in the fuzzy controller.

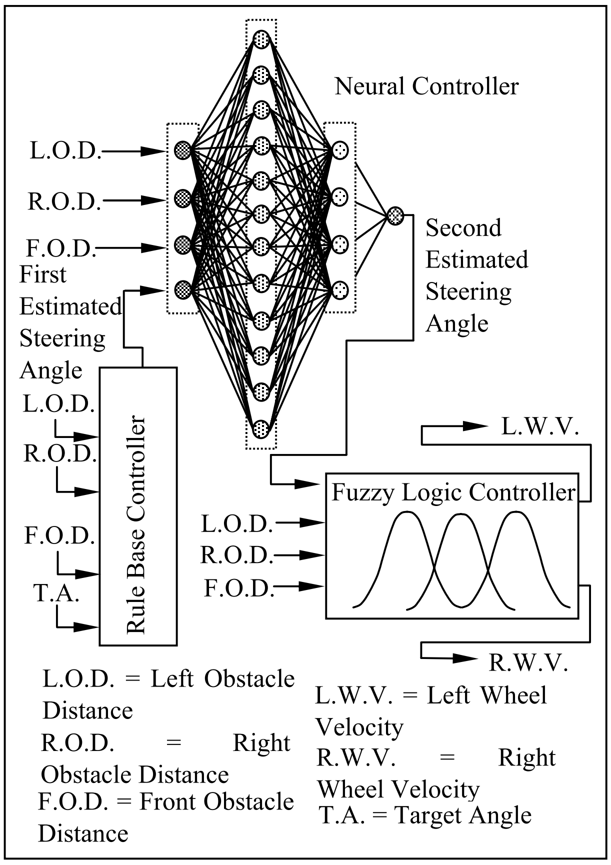

Neural Controller
L.O.D.
R.O.D.
F.O.D.
First Estimated Steering Angle
Second Estimated Steering Angle
L.O.D.
R.O.D.
F.O.D.
T.A.
Rule Base Controller
L.W.V.
Fuzzy Logic Controller
L.O.D.
R.O.D.
F.O.D.
R.W.V.
L.O.D. = Left Obstacle Distance
R.O.D. = Right Obstacle Distance
F.O.D. = Front Obstacle Distance
L.W.V. = Left Wheel Velocity
R.W.V. = Right Wheel Velocity
T.A. = Target Angle

Fig. 4 Rule-base-neuro-fuzzy technique for navigation of mobile robots

V. Demonstrations

A. Simulation

The rule-based technique has been implemented in different simulated environments. Simulations were conducted using the Window-based simulation software package 'ROBPATH' (Appendix-A) developed by the author for robot navigation. The environment generated artificially containing static obstacles as well as static targets.

B. Rule-Based-Neuro-Fuzzy Technique

1) Escape from Dead Ends

Fig. 5 and Fig. 6 shows the ability of the robots, which are trapped within dead ends, able to escape from those dead ends and find the target. Fig. 5 shows the situation at the beginning of the exercise. Ten robots are involved in the exercise. Two of the obstacles are U-shaped and two obstacles are rectangular shape representing a dead end. From Fig. 6, it can be seen that all the robots that have stayed inside have escape and able to negotiate with dead ends and find the target successfully.

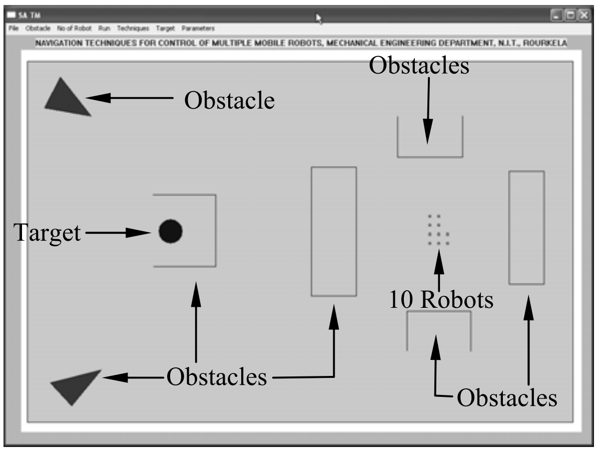

NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Obstacles
Obstacle
Target
10 Robots
Obstacles
Obstacles

Fig. 5 Escape from dead ends by fifteen mobile robots using rule-based- neuro- fuzzy technique. (Initial State)

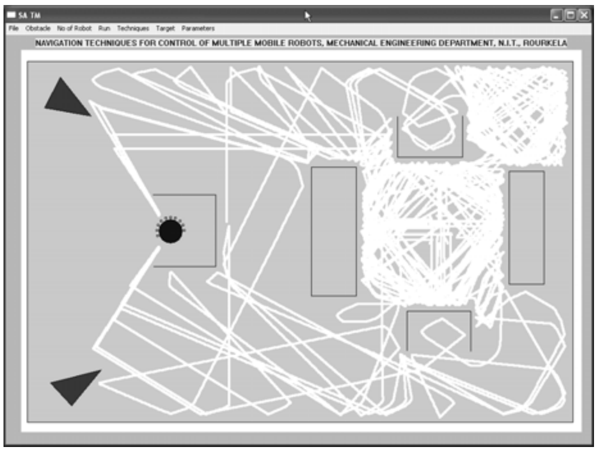

NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA

Fig. 6 Escape from dead ends by fifteen mobile robots using rule-based-neuro- fuzzy technique. (Final State)

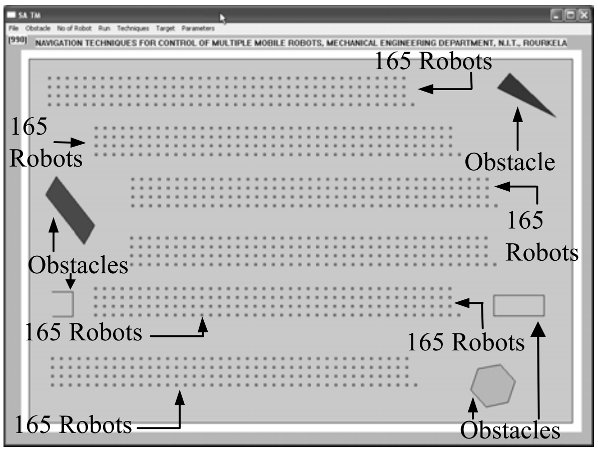

NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
165 Robots
165 Robots
Obstacle
165 Robots
Obstacles
165 Robots
165 Robots
165 Robots
Obstacles

Fig. 7 Navigation scenario of nine hundred ninety mobile robots using rule-based-neuro- fuzzy technique. (Initial State)

2) Navigation of several mobile robots

Obstacle avoidance by 'nine hundred ninety' robots using rule-based-neuro-fuzzy technique is shown in Fig. 7 and Fig. 8. Fig. 7 depicts the state at the beginning of the exercise where as Fig. 8 shows the situation some time after the exercise has begun. It can be seen that the robots are stay well away from the other robots as well as from the obstacles.

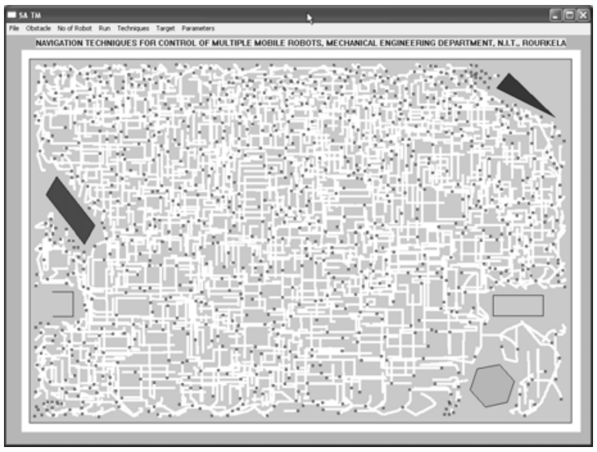

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA

Fig. 8 Navigation of nine hundred ninety mobile robots using rule-based-neuro- fuzzy technique. (Intermediate State)

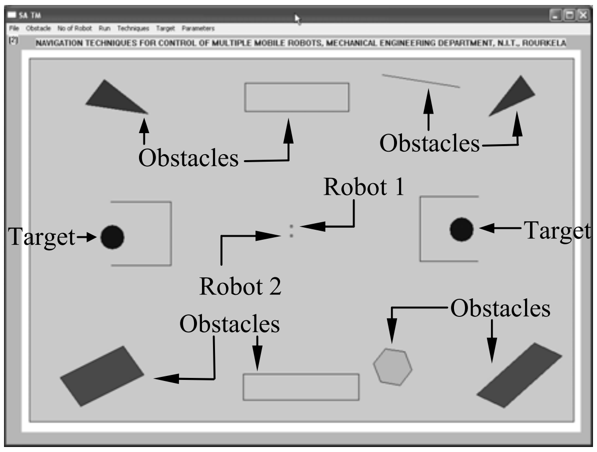

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Obstacles
Obstacles
Robot 1
Target
Target
Robot 2
Obstacles
Obstacles

Fig. 9 Collision avoidance by two mobile robots using rule-based-neuro- fuzzy technique. (Initial State)

3) Inter Robot Collision Avoidance

This exercise relates to a problem designed to demonstrate that, the robots do not collide with each other even in a highly cluttered ambience. Fig. 9 depicts the beginning of the exercise where as Fig. 10 shows the trajectories of the robots for the rule-based-neuro-fuzzy controller. It can be noted that the robots are able to resolve conflict and avoid one another and reach the target successfully.

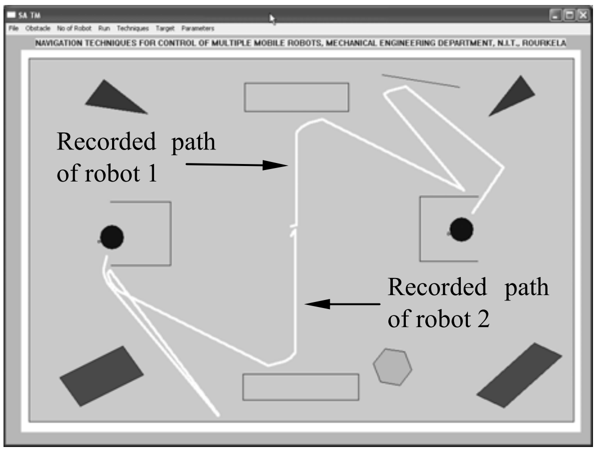

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Recorded path of robot 1
Recorded path of robot 2

Fig. 10 Collision avoidance between two mobile robots using rule-based-neuro- fuzzy technique. (Final State)

VI. Comparison between Different Controllers

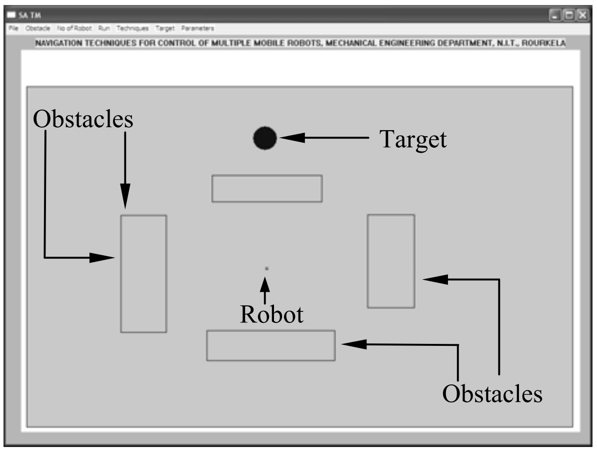

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Obstacles
Target
Robot
Obstacles

Fig. 11 Environment for one robot and one target

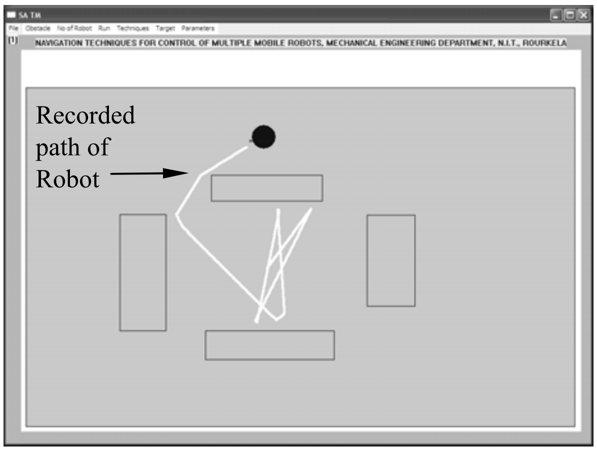

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Recorded path of Robot

Fig. 12 Navigation path for one mobile robot to reach target using rule-based controller

In this paper, work is also being conducted to determine the most appropriate technique for multiple mobile robots control out of rule-based technique and rule-based-neuro-fuzzy technique. An exercise has been devised to compare the performances of the rule-based technique, and the rule-based-neuro-fuzzy technique. In all the exercises one robot is located inside four obstacles and one target hidden from the robot view is present. Fig. 11 represents the initial condition of the environment for all the controllers. Fig. 12 and Fig. 13 depict the path traced by the robot using rule-based technique and rule-based-neuro-fuzzy techniques respectively. Total path lengths using rule-based technique and rule-based-neuro-fuzzy technique are measured in pixels for four, eight, ten, sixteen, twenty-four, forty, fifty and seventy robots. The path lengths are taken statistically from one thousand simulation results. The final results are given in Table 1. Similarly time taken to reach the target using rule-based technique and rule-based-neuro-fuzzy technique are measured for the same number of robots using statistical method. The results are given in Table 2. The path lengths and search times is giving an objective measure of the performance of the different controller. It can be noted that the rule-based-neuro-fuzzy controller performs the best.

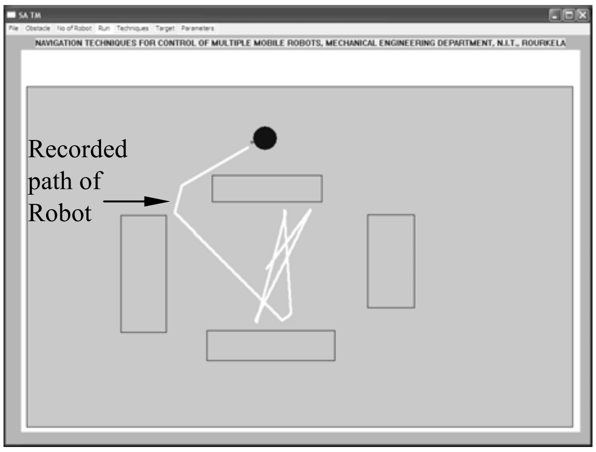

SA TM
File Obstacle No of Robot Run Techniques Target Parameters
NAVIGATION TECHNIQUES FOR CONTROL OF MULTIPLE MOBILE ROBOTS, MECHANICAL ENGINEERING DEPARTMENT, N.I.T., ROURKELA
Recorded path of Robot

Fig. 13 Navigation path for one mobile robot to reach target using rule-based-neuro-fuzzy controller

TABLE I
PATH LENGTHS USING DIFFERENT TECHNIQUES.

Number of robots	Total path length in pixels using rule-based technique	Total path length in pixels using rule-based-neuro-fuzzy technique
4	358	321
8	989	834
10	1256	645
16	1878	1723
24	2998	2534
40	4511	4105
50	5378	4002
70	11102	10303

TABLE II
TIME TAKEN TO REACH THE TARGET USING DIFFERENT TECHNIQUES.

Number of robots	Total time taken in seconds using rule-based technique	Total time taken in seconds using rule-based-neuro-fuzzy technique
4	12.54	9.48
8	24.45	18.56
10	26.23	19.36
16	36.36	28.49
24	55.09	49.05
40	87.21	69.05
50	127.57	99.45
70	265.01	245.54

VII. CONCLUSION

The present investigation has described techniques for controlling the navigation of multiple mobile robots using rule-based technique and rule-based-neuro-fuzzy technique in a highly cluttered environment. The rule-based technique has a set of rules obtained through rule induction and subsequently with manually derived heuristics. This technique employs rules and takes into account the distances of the obstacles around the robots and the bearing of the targets in order to compute the change required in steering angle. With the use of Petri net model the robots are capable of negotiate with each other. It has been seen that, by using rule-based-neuro-fuzzy technique the robots are able to avoid any obstacles (static and moving obstacles), escape from dead ends, and find targets in a highly cluttered environments. Using these techniques as many as one thousand mobile robots can navigate successfully neither colliding with each other nor colliding with obstacles present in the environment. It was observed that the rule-based-neuro-fuzzy technique is the best compared to the rule-based technique for navigation of multiple mobile robots.

APPENDIX – A

SOFTWARE USED FOR ROBOT NAVIGATION:

The 'ROBPATH' used for navigation demonstrations reported in this paper is developed by the author. The software runs on a PC operating under WINDOWS NT/95/98/2000/XP. The menus incorporated in the software are described below.

Obstacle Menu:

The Obstacle Menu allows the user to draw different types of obstacles in the robots' environment. The obstacles that can be constructed are shown in Fig. 14.

Number of robot Menu:

Using this menu, a user can draw any number of robots (in between 1 to 1000) as required to be placed in the environment. For example fifteen robots are shown in Fig. 15.

Run Menu:

With this menu, the user can choose to run the software in simulation mode or control the navigation of real mobile robots. The menu is given in Fig. 16.

Techniques Menu:

This menu enables the user to select the techniques to control the navigation of the robots. The menu is shown in Fig. 16.

Target Menu:

This menu is for placing targets in the environment. Any number of targets can be Chosen for the robot environment (Fig. 16).

Manual Command (Parameter Menu)

This menu contains four commands (Fig. 16).

a) Start_Again: This command is for re-starting a process with exiting software.

b) Refresh_Screen: This command is for cleaning the screen. It has the same action as clicking the right button on the mouse.

c) Robot_Behaviour: This command activates a dialog box that enables the user to select a particular robot and control its movements manually. The dialog box can be seen in Fig. 16.

d) Drag_Obs: Using this sub-menu the user can drag any obstacle to any place in the environment at any time to re-arrange the cluttered environment.

Obs_Delete: By the help of this menu the user can delete any obstacle at any time from the environment.

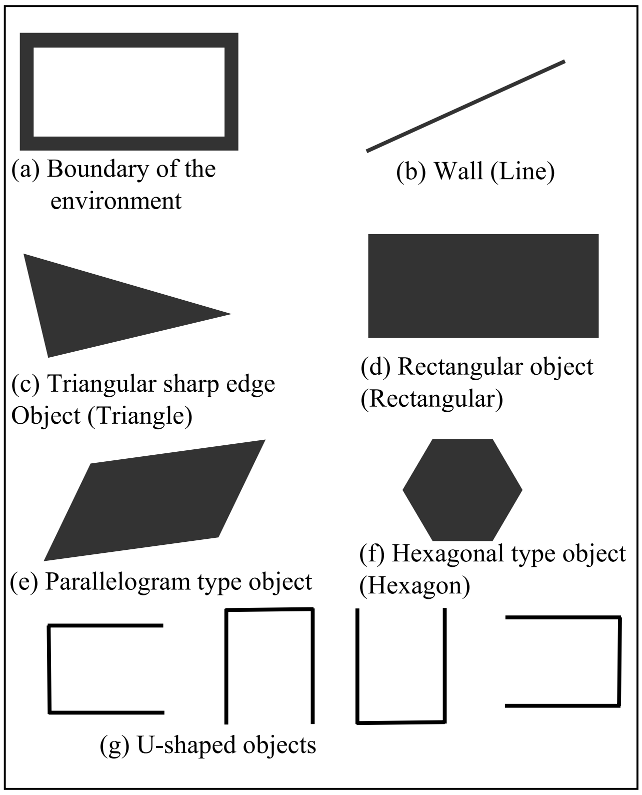

(a) Boundary of the environment
(b) Wall (Line)
(c) Triangular sharp edge Object (Triangle)
(d) Rectangular object (Rectangular)
(e) Parallelogram type object
(f) Hexagonal type object (Hexagon)
(g) U-shaped objects

Fig. 14 Obstacles

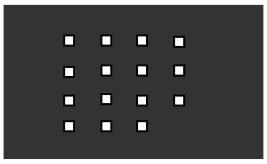


Fig. 15 Fifteen mobile robots

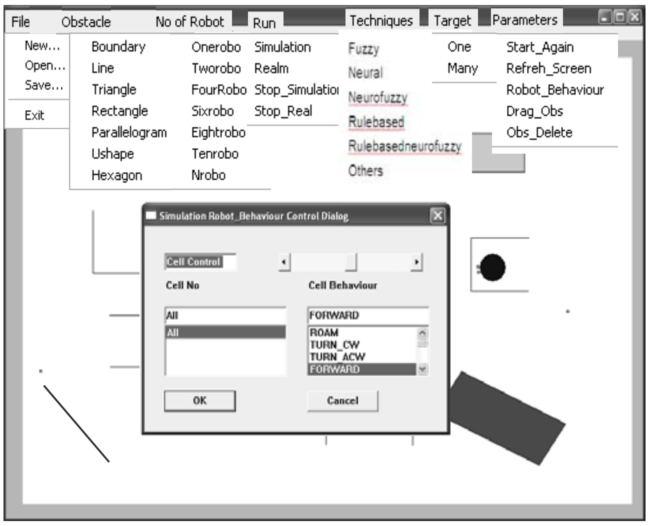


Fig. 16 View of the software (ROBOPATH) front-end user for navigation of multiple mobile robots

APPENDIX – B

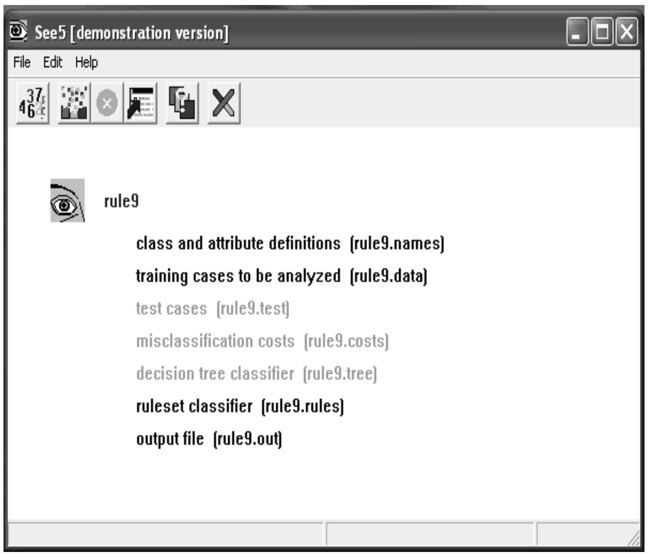
DATA MINING TOOLS SEE5

Data mining is about extracting patterns from a warehoused data. See5 is data mining software. Some important features of See5 (Fig. 17 and Fig. 18) are described below:

See5 has been designed to analyse substantial databases containing thousands to hundreds of thousands of records and tens to hundreds of numeric or nominal fields.

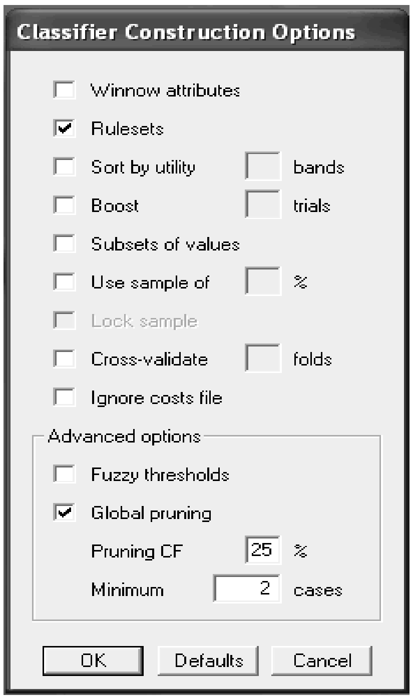
To maximize interpretability, See5 classifiers are expressed as decision trees or sets of if-then rules, forms that are generally easier to understand than neural networks.

See5 is easy to use and does not presume advanced knowledge of Statistics or Machine Learning.


See5 [demonstration version]
File Edit Help

rule9
class and attribute definitions (rule9.names)
training cases to be analyzed (rule9.data)
test cases (rule9.test)
misclassification costs (rule9.costs)
decision tree classifier (rule9.tree)
ruleset classifier (rule9.rules)
output file (rule9.out)

Fig. 17 The main window of See5


Classifier Construction Options

- [ ] Winnow attributes
- [✓] Rulesets
- [ ] Sort by utility ___ bands
- [ ] Boost ___ trials
- [ ] Subsets of values
- [ ] Use sample of ___ %
- [ ] Lock sample
- [ ] Cross-validate ___ folds
- [ ] Ignore costs file

Advanced options
- [ ] Fuzzy thresholds
- [✓] Global pruning
- Pruning CF 25 %
- Minimum 2 cases

OK Defaults Cancel

Fig. 18 Main dialog box

APPENDIX – C

NEURAL NETWORK

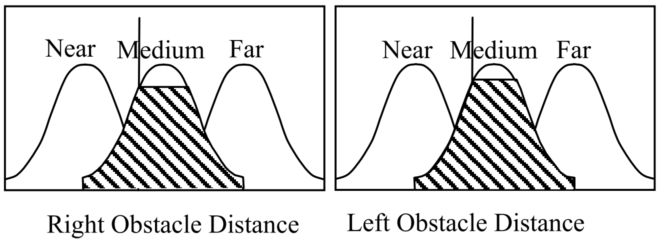
The neural network used in this paper is a multilayer perceptron having back-propagation algorithm [16]. The number of layers selected here is found empirically to facilitate training of the neural network. The input layer has four neurons. Out of the four neurons three are meant to receive the distances of the obstacles from front, left and right, where as the fourth neuron is for the target angle. If no target is detected in the environment, the input to the fourth neuron is "zero". The first hidden layer has twelve neurons and the second hidden layer has four neurons. The numbers of neurons in the hidden layers are found empirically. The output layer has a single neuron meant to produce the steering angle to control the direction of movement of robots. Fig. 4 depicts the neural network highlighting the details of the neurons with its input and output signals.

APPENDIX – D

Fuzzy Logic Controller (FLC) [17] used in rule-based-neuro-fuzzy technique has been discussed briefly below. Navigation of multiple mobile robots in presence of static and moving obstacles can be carried out by specifying a set of fuzzy rules taking into account the different environmental situations. For this purposes the input to the FLC are left obstacle distance, right obstacle distance, front obstacle distance and target angle. The output from FLC is left wheel velocity and right wheel velocity of mobile robots. The fuzzy rules help the robots to avoid obstacles and find targets.

Gaussian membership function is used in the fuzzy logic controller. Linguistic variables such as "far", "medium" and "near" are taken for Gaussian membership function for navigation of multiple mobile robots. Linguistic variables such as "positive", "zero" and "negative" are defined for the target angle with respect to target. Terms like "slow", "medium", and "fast" are considered for left wheel velocity and right wheel velocity of mobile robots. Some of the fuzzy control rules are activated according to the information acquired by the robots. The outputs of the activated rules are weighted by fuzzy reasoning and the velocities of the rear driving wheels of the robots are calculated. A typical situation is explained in Fig. 19 to calculate the velocity of right wheel and left wheel. Two rules are considered for calculation of driving wheel velocity. First rule is about a situation when the robot is located in an environment where right obstacle distance is 'medium', left obstacle distance is 'near', front obstacle distance is 'far' and the target angle is 'zero'. For the above condition the right wheel velocity should 'medium' and left wheel velocity should 'fast' (i.e., to turn right).

Similarly second rule is applied when the robot is located in an environment where right obstacle distance is 'near', left obstacle distance is 'near', front obstacle distance is 'far' and the target angle is 'zero'. For this situation the right wheel velocity should 'slow' and left wheel velocity should 'slow' so that the robot moves straight. The resultant of right wheel and left wheel velocity is shown Fig. 20.



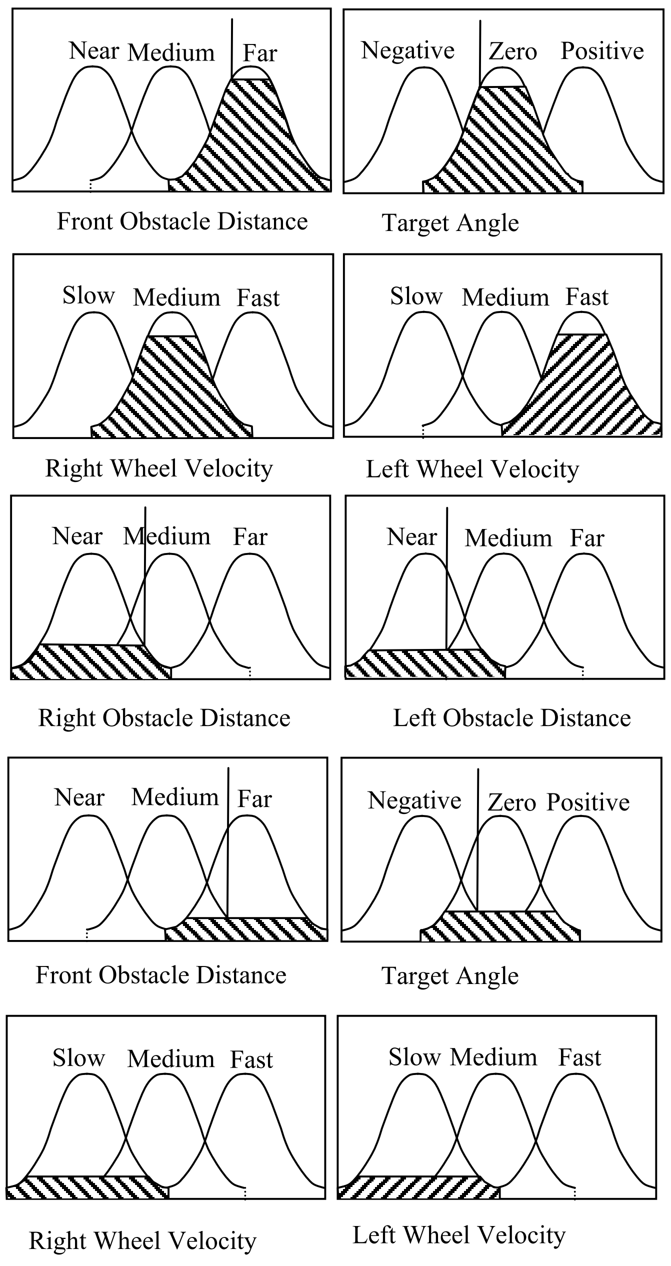
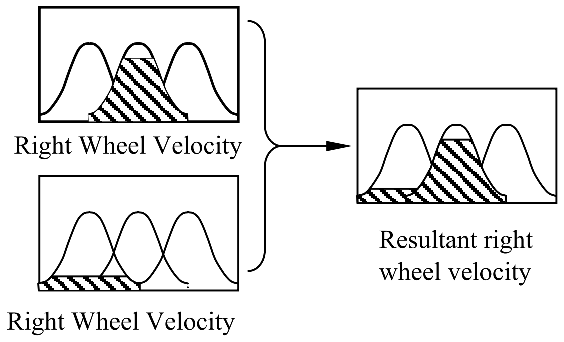


Fig. 19 Left and right velocity when fuzzy rules are activated



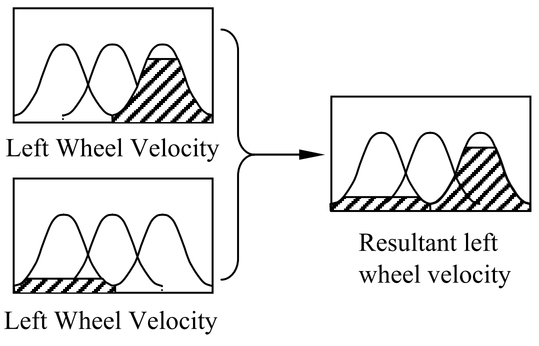


Fig. 20 Resultant left and right velocity

References

[1] N. T. Gürman, “The neural network model RuleNet and its application to mobile robot navigation,” *Fuzzy Sets and Systems*, Volume 85, Issue 2, 23 January 1997, pp. 287-303.

[2] W. Li, C. Ma, F. M. Wahl, “A neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments, “ *Fuzzy Sets and Systems*, Volume 87, Issue 2, 16 April 1997, pp. 133-140.

[3] C. W. Barfoot, M. Y. Ibrahim, “Development of an adaptive fuzzy behavioural control system with experimental and industrial applications,” *Computers & Industrial Engineering*, Volume 34, Issue 4, September 1998, pp. 807-811.

[4] Jelena and Nigel, “Neuro-fuzzy control of a mobile robot,” *Neurocomputing*, Volume 28, Issues 1-3, October 1999, pp. 127-143.

[5] L. Acosta, G.N. Marichal, L. Moreno, J. A. Méndez, J.J. Rodrigo, “Obstacle Avoidance Using the Human Operator Experience for a Mobile Robot,” *Journal of Intelligent and Robotic Systems*, April 2000, Volume 27, No. 4, pp. 305-319.

[6] G.N. Marichal, L. Acosta, L. Moreno, J.A. Mendez, J.J. Rodrigo, M. Sigut, “Obstacle avoidance for a mobile robot: A neuro-fuzzy approach,” *Fuzzy Sets and Systems*, 1 December 2001, Volume 124, No. 2, pp. 171-179.

[7] K. Althoefer B. Krekelberg, D. Husmeier, L. Seneviratne , “Reinforcement learning in a rule-based navigator for robotic manipulators,” *Neurocomputing*, Volume 37, Issues 1-4 , April 2001, pp. 51-70.

[8] S. Nefti, M. Oussalah, K. Djouani, J. Pontnau, “Intelligent Adaptive Mobile Robot Navigation,” *Journal of Intelligent and Robotic Systems*, April 2001, Volume 30, No. 4, pp. 311-329.

[9] E. Tunstel, A. Howard, H. Seraji, “Rule-based reasoning and neural network perception for safe off-road robot mobility,” *Expert Systems*, September 2002, Volume 19, No. 4, pp. 191-200.

[10] M. A. F. de Souza, M. A. G. V. Ferreira, “Designing reusable rule-based architectures with design patterns,” *Expert Systems with Applications*, Volume 23, Issue 4, November 2002, pp. 395-403.

[11] Y-K. Na and S-Y. Oh, “Hybrid Control for Autonomous Mobile Robot Navigation Using Neural Network Based Behavior Modules and Environment Classification,” *Autonomous Robots*, September 2003, Volume 15, No. 2, pp. 193-206(14).

[12] J. Dietrich, A. Kozlenkov, M. Schroeder, G. Wagner, “.Rule-based agents for the semantic web,” *Electronic Commerce Research and Applications*, Volume 2, Issue 4, Winter 2003, pp. 323-338.

[13] B. S. McIntosh, R. I. Muetzelfeldt, C. J. Legg, S. Mazzoleni, P. Csontos, “Reasoning with direction and rate of change in vegetation state transition modeling,” *Environmental Modelling & Software*, Volume 18, Issue 10, December 2003, pp. 915-927.

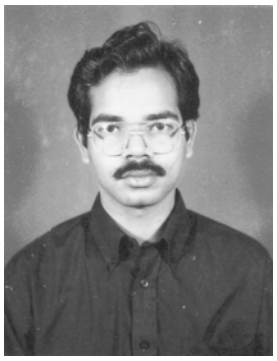
[14] Clementine Software, Version-5, Available : http://www.spss.com/Clementine/, 2000.

[15] J.L. Peterson, *Petri Net theory and the Modelling of Systems* (Prentice-Hall, Englewood Cliff.N.J., 1981).

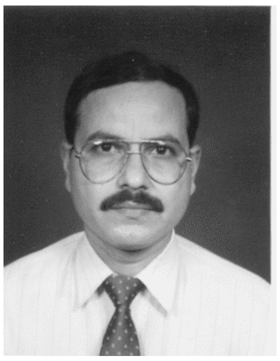
[16] D.T. Pham, D. R. Parhi, “Navigation of multiple mobile robots using a neural network and a Petri net model,” *Robotica*, U.K., Volume 21, 2003, pp. 79-93

[17] S. K. Pradhan, D. R. Parhi, A. K. Panda, “Neuro-fuzzy techniques for navigation of multiple mobile robots,” *Fuzzy optimization and decision making*, to be published.

S. K. Pradhan is Visiting Faculty of Mechanical Engineering with the N.I.T., Hamirpur, HP, India and doing research in the field of Mobile Robots Navigation



D. R. Parhi had achieved his first Ph.D. in Mobile Robotics from Cardiff School of Engineering, UK and second Ph. D. in Vibration Analysis of Cracked structures from India. Currently he is now engaged in Mobile robot Navigation research.



A. K. Panda earned B.Sc. Engg. in Electrical engineering from Sambalpur University in the year 1987 and Ph.D. in Power Electronics from Utkal University in 2001. Currently Mr. Panda’s research interest is in the field of Artificial Intelligence and Robotics.