Proceedings of the 6th WSEAS Int. Conf. on FUZZY SYSTEMS, Lisbon, Portugal, June 16-18, 2005 (pp79-82)

New intelligent controller for mobile robot navigation in unknown environments

SALVATORE PENNACCHIO, FRANCESCA ABISSI,
SABINA PETRALIA, GIANNANTONIO STENDARDO
Università degli Studi di Palermo
Viale delle Scienze, 90128 Palermo
ITALY

Abstract: - This paper presents, the FU.LO.RO. (FUzzy LOgic RObot), a new controller for mobile robot, which can moves itself autonomously in an unknown environments. This system has been developed following two different approaches: the first one enables a fuzzy controller to determine the robot's behaviour using fuzzy basic rules; the second one uses a neurofuzzy controller.

Key-Words: - Mobile Robot, Fuzzy Logic, Neural Network, Fuzzy Network, Back Propagation, Controller.

1 Introduction

Designing a totally autonomous robot implies several aspects such as path planning, sensor-motor control, obstacle detection and obstacle avoidance. The traditional computational methods hadn't have many success to design a dynamical properties and real-time requirements of robot control; so, new neural network and fuzzy logic techniques were born to resolve the opened problems.

In the last years, many solutions were proposed to design robotic systems. For example, neural networks are used to back up a simulated truck [1], fuzzy logic controllers are used to control a model race car [2, 3, 4]; and also, many controllers were designed to manage the movement of a robot to goal in environments where there are fixed and mobile obstacle[5]; to manage the movement across a rough terrain getting information about slope, irregularity and unevenness [6, 7]; to control and to operate on fluid environment [8, 9, 10, 11, 12, 13].

This paper, introduces a new controller, called FU.LU.RO. (FUzzy LOgic RObot), for mobile robots in unknown environments. This systems enables a circular robot moves autonomously in an unknown map. The robot is equipped with three proximity sensors which detect north side obstacles, east side obstacles and west side obstacles. The information detected from sensors and the robot's current speed are used to determine the changes of direction and acceleration. Comparing two different approaches is the basic idea: the first one lets a fuzzy controller to determine the robot's behaviour using fuzzy basic rules; the second one uses a neurofuzzy controller trained in two different way. In the first a fuzzy controller gives a training set to a neural network; in the second the software ANFIS has been used.

The development and simulation tool is Matlab® 6.5 R13 of Mathworks inc. [14].

We have designed a visual simulator in Microsoft Visual Basic 6.0 to enable to monitor robot's dynamic. The communication between Visual Basic and Matlab is possible using Matlab as local server.

2 General Approaches

This paper proposes the compare of two different approaches for the robot's navigation in unknown environments.

The first solution uses a fuzzy logic controller [16] that determines the robot's behaviour with fuzzy basic rules. The inputs of the controller are the signals of the robot's sensors and current speed. The general architecture of our system is shown in Fig.1:

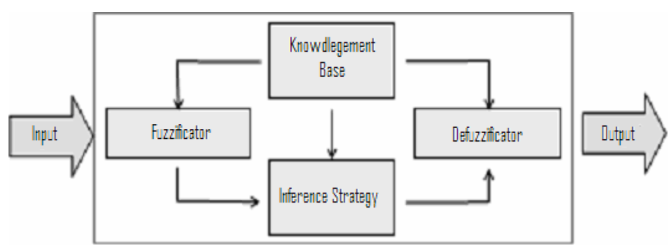

Fig.1: Fuzzy Controller

A fuzzificator turns the inputs in an membership degree with a linguistic variable. The knowledge base is composed by two parts:

1. some membership functions' parameter of input and output variables. These parameters allow the input fuzzification and then output defuzzification.
2. a basic rules set containing if-then rules which computes the fuzzificated input to obtain some appropriate output values.

The inference strategy compute the right output: every fuzzy rule is processed to obtain a fuzzy variable. An output fuzzificated is obtained using the OR operator, then this output is turned by the defuzzificator in a numeric value.

A better solution can be found using together a neural network and a fuzzy controller [17].

Usually, the neural network is trained using a supervisionated algorithm. The network learns the relationship between inputs and outputs. This approach has the disadvantage of a big training set for preparing the robot to several situations in the real world.

Comparing two Neurofuzzy system permits to design different approaches. The first one uses the ANFIS tool using backpropagation algorithm [18].

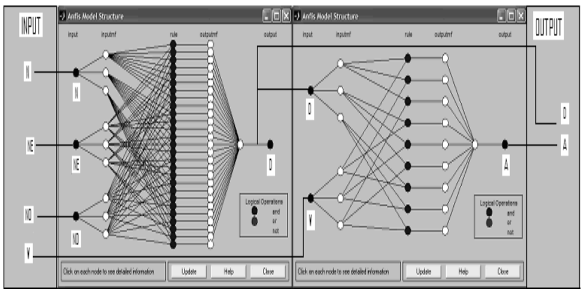

Fig.2: Neurofuzzy System trained with the ANFIS tool

The second solution uses a fuzzy controller which provides a training set to a neural network with a back propagation algorithm. This approach enables the robot to move in an unknown environment for avoiding obstacles. The general architecture of this system is shown in Fig.3:

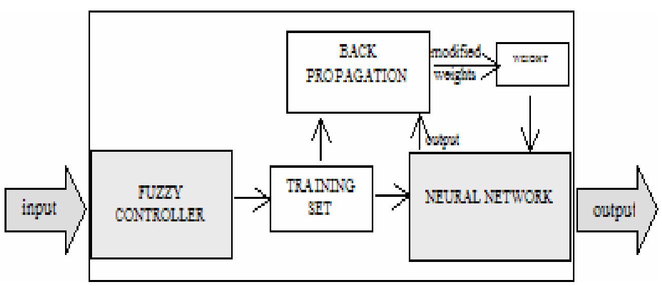

Fig.3: Neuro-Fuzzy Architecture

The sensors data and the robot's current speed are the inputs of the fuzzy controller. These inputs are processed by controller and they generate the desired robot's behaviour. The input/output couples obtained formed the training set that is the input of neural network. This uses the back propagation algorithm to improve the solution obtained using the fuzzy controller only.

3 The fuzzy controller

The fuzzy controllers simplify the project of a control system when there is an "a priori" knowledge of the system, also if there aren't exact methods to calculate the rules number, the variables, the membership functions type and the parameters and the methods to choose the operators. Our fuzzy controller uses four inputs and two outputs. We suppose the robot has three proximity sensors that give information about obstacle presences across north (0 degrees), north-east (45 degrees) and north-west (-45 degrees) directions. The robot detects the obstacles until a distance of 12 metres. So, inputs of controller are: the distance across the three directions of eventual obstacles and robot's current speed. The inputs are numeric values of fixed ranges and reflect as possible as the real world. The outputs of fuzzy controller are acceleration and the new robot's direction. Acceleration can be positive, null or negative; the direction means the angle's variation.

The design of a fuzzy controller is composed by three important characteristics: the definition of linguistic variables; the choice of relationships between variables across the use of fuzzy rules (if-then); the choice of the defuzzification heuristic.

The input linguistic variables are the controller inputs. Three fuzzy set has been defined for each of these inputs; in particular, the distance linguistic variables , the fuzzy sets are : *near (DV), medium (DM), far (DL);* about speed, the fuzzy sets are: *low (VB), medium (VM), high (VA).*

Every fuzzy set is characterized by a membership function that associates a membership degree to relative set with a values of distance or speed. For example, as shown in Fig.4, an object, far 2 metres across north direction, has a membership degree equal to 1 for the *near* fuzzy set, equal to 0 for the other fuzzy sets.

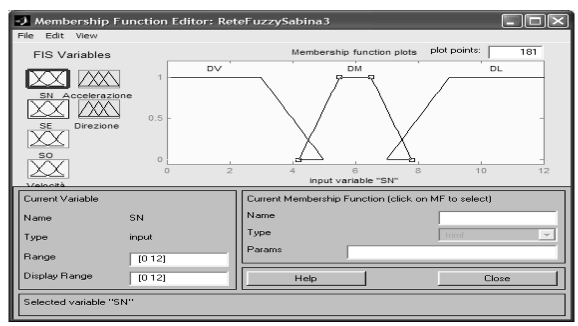

Fig.4: Fuzzy Set and MF of SN

The output linguistic variables are: the acceleration and the direction. Four fuzzy sets are defined for acceleration: *braking (FR), negative (NG), null (NU) and positive (PS)*; about the direction we have defined five fuzzy sets: *east U-turn (IE), north-east (NE), north (N), north-west (NO), west U-turn (IO).* So, four MF for acceleration and five for direction have been chosen. Knowledge base is composed by eleven rules if-then where the precedent part values are combined with the intersection operation using the minimum operator AND:

1. If (Speed is VB) then (Acceleration is PS);
2. If (Speed is VM) then (Acceleration is NU);
3. If (Speed is VA) then (Acceleration is NG);
4. If (SN is DV) and (SE is DV) and (SO is not DV) then (Acceleration is FR)(Direction is IO);
5. If (SN is DV) and (SE is not DV) and (SO is DV) then (Acceleration is FR)(Direction is IE);

6. If (SN is DV) and (SE is DV) and (SO is DV) then (Acceleration is FR)(Direction is IE);
7. If (SN is not DV) and (SE is DV) and (SO is not DV) (Direction is NO);
8. If (SN is not DV) and (SE is not DV) and (SO is DV) (Direction is NE);
9. If (SN is DV) and (SE is DV) and (SO is not DV) and (Speed is not VB) then (Acceleration is NG)(Direction is IO);
10. If (SN is not DL) and (SE is not DV) and (SO is DV) and (Speed is not VB) then (Acceleration is NG)(Direction is IE);
11. If (SN is DV) and (SE is not DV) and (SO is not DV) and (Speed is not VB) then (Acceleration is NG)(Direction is IE);

The successive step consists in evaluating the rule consequent part trough the intersection operation and cutting the output fuzzy set with the value obtained from the rule antecedent part. The model of the controller used is Mandani, so the output value is obtained applying the maximum operator OR to every rule outputs. In Fig.5 is shown the inference between different rules:

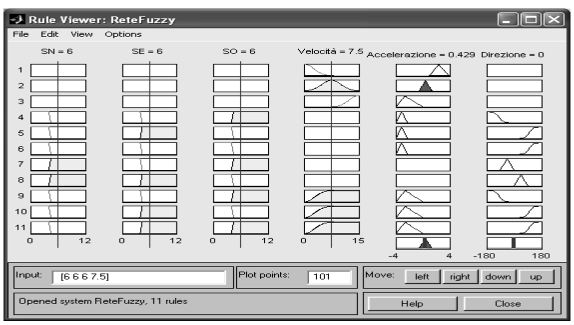

Fig.5: Inference between Rules

The centroid method is applied for defuzzification: the area's barycentre is calculated and output is the value of the abscissa's barycentre.

4 Fuzzy Network

In this paper two methods for training networks are described. The first one uses knowledge methods, which allow to modify some parameters of the fuzzy controller for minimizing the error. For this kind of training we used the ANFIS (Adaptive-Network-Based-Fuzzy-Inference-System) tool, considering the fuzzy controller like a particularly neural network. The training set for the ANFIS is generated with a supervisionated algorithm likely for a simple neural network. The goal of this training is to minimize medium square error. This method had minimized the error in 1395 epochs, using 600 point of Input/Output as shown in Fig. 6

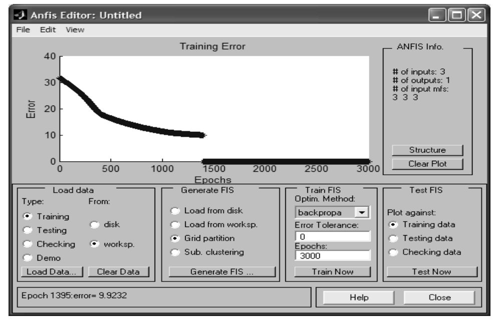

Fig.6: ANFIS Network Training

The second method is composed by two phases: the design of a fuzzy controller and then the design of a neural network with back propagation algorithm; the training set of the neural network is composed by the input/output couples provided by the fuzzy controller. The fuzzy controller used in this context is that one described before. The controller produces some outputs for an input value set, the first part of this set is produced randomly the second one is produce to improve the behaviour in the obstacle absence. Before the input/output couples provided by the fuzzy controller are used like training set for the network's training and then like validation set to verify the capacity of network's generalization. The neural network's input are four like those ones of the fuzzy controller: the distance across the three direction (north, north-east, north-west) and the speed of the robot. The outputs are the acceleration and the direction. A network has been chosen with forward connections with 7 hidden units. The membership function used is the hyperbolic tangent sigmoid which inputs can change in every the real axis; the outputs are normalized in [-1 1]. So the output must be multiplied by a fixed factors. As in Fig. 7, the network minimized the error in 100 epochs, using 500 pints of input/outputs and obtaining a precision of 0.0036.

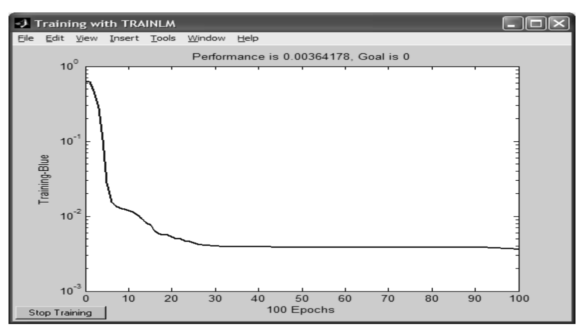

Fig 7: Neuro-Fuzzy Network Training

5 Visual Simulator

A visual simulator was designed for allowing to view robot's dynamic; this simulator was realized with Microsoft Visual Basic 6.0. Matlab was used as local serve to allow the communication between Visual Basic e Matlab® 6.5 R13. In this simulator user can chooses which controller must be used for the behaviour of robot, and user can set the initial robot's position, the simulation speed and the sensors' errors; also user can draw new obstacles or walls in the map. In Fig. 8 is shown the simulator for the case in which a Neuro-Fuzzy network cause the robot's behaviour.

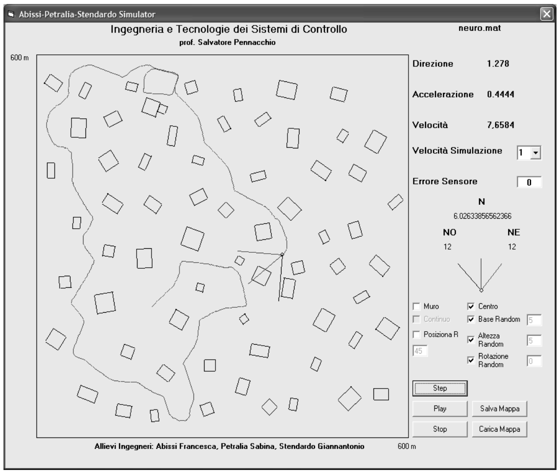

Fig. 8: APS-Simulator

6 Application

The controller designed in this paper can be used to rebuild the map for hostile surroundings. For example if the user want to rebuild the map for a camp with landmines, user can equip the robot with appropriate sensors that can communicate with graphic device, at the end of the exploration with the rebuilding of the map user can cross the camp.

7 Conclusions

From the experimental results, the behaviour of both Neurofuzzy controllers is definitely better then the fuzzy controller behaviour. Besides, the ANFIS controller generates a robot's behaviour that reflects less the real behaviour . The Neuro-Fuzzy controller is simpler to design because it doesn't need a good quality fuzzy controller thanks to the combination with the neural network.

References:

[1] D.Nguyen and B.Widrow, "The truck backer-upper: An example of self learning in neural networks", *Proc. of the Int. Joint Conference on Neural Networks*, Vol. 2, pp. 357-364, 1989.

[2] C.Von Altrock, B.Krause and H.J.Zimmermann, "Advanced Fuzzy logic control of a model car in extreme situation*", Fuzzy Sets and System*, 48(1):41-52, 1992.

[3] M.Maeda, Y.Maeda and S.Muratami, "Fuzzy Drive control of an autonomous mobile robot", Fuzzy Sets and System, 39: 195-204, 1991.

[4] M.Sugeno, T.Murofushi, T.Mori, T.Tatematsu and J.Tanaka, "Fuzzy algorithmic control of a model car by oral instruction", *Fuzzy Sets and System*, 32: 207-219, 1989.

[5] H.Kovama et.al, "Study of obstacle Avoidance problem for Mobile Robot Using Fuzzy Production System", J.*of the Robotics Society of Japane*, Vol.9, 1991.

[6] A.Kelly and A.Stendz, "Rough terrain autonomous mobility", *J.Auton Robots*, Vol.5, no.2, pp.129-198, 1998.

[7] K.Iagnemma, F.Genot and S.Dubowsky, "Rapid phisics-based rough terrain rover planning with sensor and control uncertainty", in *Proc.IEEE Int.Conf.Robotics and Automation,* Vol.3, Detroit, MI, 1999, pp.2286-2291.

[8] S. Pennacchio, F.M. Raimondi, "A Neuro Fuzzy Control System for a Biped Robot", *WSEAS Transactions on Systems*, Issue 10, Vol. 3, December 2004, pp. 3152-3156.

[9] S. Pennacchio, F.M. Raimondi, "Fuzzy Motor Schema for Mobile Robots", WSEAS Transactionson Systems, Issue 5, Volume 3, July 2004, pp.2211-2214.

[10] B.Levesque and M.J.Richard, "Dinamic analisys of a manipulator in a fluid environment", *Int.J.Robot.Res.,* Vol.13, no.3, pp.221-231, 1994.

[11] S.McMillan, D.E.Orin and R.B.McGhee, "Efficient, dynamic simulation of an underwater vehicle with a robotic manipulator", *IEEE Trans.Syist., Man, Cybern*., Vol.25, pp.1194-1206, Aug. 1995.

[12] K.P.Goheen and E.R. Jefferys, "On adaptive control of remotely operated underwater vehicles", *Int. J. Adaptive Contr. Signal Processing,* Vol.4, pp.287-297,1990.

[13] H. S. Choi, "Modeling of robot manipulators working under the sea and the design of a robust controller", *Robotica,* Vol.14, pp.213-218, 1996.

[14] The Student Edition of Matlab-The language of technical computing, the Matworks.

[15] H.J.Zimmermann, "Fuzzy Set Theory- and its Applications*", Kluwer Academic*, Boston, 1991.

[16] C.Lin and C.S.Lee, "Neural-network-based fuzzy logic control of an autonomous mobile robot", *Fuzzy Sets and System*, 39:195-204, 1992.

[17] D.E.Rumelhart, G.E.Hinton and R.J.Williams, "Learning internal representations by error propagation", In: D.E. Rumelhart and J.L.Mc Clelland (Eds), "Parallel Distributed Processing", Vol.1, pp.318-362, MIT Press, Cambridge, 1989.