A new model of fuzzy system for mobile robots in unknown environment

SALVATORE PENNACCHIO, DANilo CANDELA, FABIO CENTINEO, MASSIMO LA ROSA
University of Palermo
Viale delle Scienze
ITALY
http://www.dias.unipa.it

Abstract: This paper introduces a new model of control system for mobile robot navigation in unknown environments. This model follows two approach, the first one is based on fuzzy logic and the second one is based on neuro fuzzy. Both approaches are directed to generate a collision free path avoiding static obstacles. This paper presents a possible optimization skill for the fuzzy controller designed on genetic algorithm.

Key-Words: Fuzzy, Neuro Fuzzy, Genetic Algorithm, Inference

1 Introduction
Robot navigation is based on the control of the motion from a start point to an end point in a workspace following a wanted path.

Control systems of robots in unknown environments have concentrated research over the past few years.[1,2,3] Autonomous navigation systems are usually classified in the following categories according to the characteristics of the environment in which they have to move:

- structured or known environment;
- semi-structured or partially known environments;
- unstructured or unknown environments.

When the robot moves in a partially or completely unknown workspace, the off-line techniques cannot be applied. Thus, the sensor-based local path planning, called obstacle avoidance is required in the navigation of mobile robots. The repulsive force [4] and the virtual force filed concepts could be applied. However, these methods have shortcomings as it is difficult to find the force coefficients of effect on the velocity and direction of mobile robots in complex workspaces that can not be described as a mathematical model[5]. Another classical way is to send the robot to discover its world and define some landmarks that can used for navigation[6]. In similar conditions, the robot relies heavily on its sensors, map making and updating. However, natural workspaces present a large amount of uncertainty, and mapping techniques are time and memory consuming techniques.

So it is necessary an approach based on fuzzy logic or based on neural network with fuzzy logic (neuro fuzzy) that deal with the uncertainty and can operate with several situations with no knowledge about models or maps of the workspace.

This paper suggests an algorithm that determine a collision free path avoiding static obstacles. The robot is equipped with a vision system based on laser scan. The algorithm subject of this paper will be implemented by an inferential fuzzy system otherwise with neuro fuzzy networks.

In the end will be show a possible optimisation of the adopted solutions by genetic programming.

2 Problem Formulation
Both implementations of the avoid obstacle algorithm, that is fuzzy and neuro fuzzy, are based on the same work’s assumptions:

a) always consider nearest obstacle: step by step robot detects an environment with only a obstacle;
b) before every decision, the robot will be direct with the direction of the target, that is it rotates of an angle $\theta_e$.

Fig. 1 Positioning of the robot toward the target

For an intelligent robot behavior to avoid obstacle be necessary consider that:

1. Bigger far is the obstacle than less will be the robot detour (little steering angle);
2. bigger is the inclusive angle between obstacle direction and target direction than less will be steering angle.

This movement is quantified on physical consideration: indicate with $\theta_l$ the angle included between target direction and obstacle direction, with $RO$ the distance between obstacle and robot. Hence the inputs to the inferential
engine will be $\theta l$ and RO. Indicate with $\theta out$ the steering angle and with Movement the movement performed by the robot; they are both outputs of the inferential engine.

In this paper it is supposed that the $\theta out$ angle in a range between -90 and 90 degree and furthermore and it is assumed the following convention:

- $\theta l$ is negative when the obstacle is on the left side of the robot (that is the robot will rotate of a positive angle $\theta out$)
- $\theta l$ is positive when the obstacle is on the right side of the robot (that is the robot will rotate of a negative angle $\theta out$)

From the controller based on inferential system we want to obtain:

- robot detour regard the target direction, represented by a steering angle;
- movement of the robot.

If no obstacle is detected by the vision system, that is the obstacles are too far, software module will provide, further the steering angle $\theta e$ of the assumption b), the movement to go toward to the target.

Basically are implemented two behaviour:

- avoid obstacle;
- move to goal.

This behaviors makes up two layer that will be activated depending on the procedure in advance:

3 Fuzzy controller

The development of techniques for autonomous navigation in real-world environments constitutes one of the major trends in the current research on robotics; a robot is autonomous when it is able to move purposefully with no help or intervention from a human user. One of the important aspects that is still deemed important to consider in mobile robot is collision avoidance. Prior knowledge about environment is in general incomplete because the environment changing and moreover uncertainty can occur in sensor data caused by environmental features. In this scenario fuzzy logic play an important role. In fact it has features that make it an adequate tool to address uncertain data, processes it and obtains certain finite data.

We want to design a fuzzy logic controller that perform a steering angle ad a movement to allow the robot to avoid obstacle. Main operation that a fuzzy inference system
have to allow are[7]:

- input fuzzification;
- fuzzy inference;
- output fuzzy composition;
- defuzzification;

Inputs of our system aren’t fuzzy but crisp values generated by the laser scan and processed by the software module so as to have the angle \( \theta l \) and the distance \( RO \). So we have to fuzzifie the inputs to the inferencial engine and defuzzifie the outputs form it. We have defined this seven fuzzy set for the input \( \theta l \) and for the output \( \theta_{out} \):

- \( NL \) : Negative Large
- \( NM \) : Negative Medium
- \( NS \) : Negative Small
- \( ZE \) : Zero
- \( PS \) : Positive Small
- \( PM \) : Positive Medium
- \( PL \) : Positive Large

The fuzzy sets for the input \( RO \) and the output \( Movement \) are:

- \( ZE \) : Zero
- \( VN \) : Very Near
- \( NE \) : Near
- \( FA \) : Far
- \( VF \) : Very Far

For the inferencial engine we have selected a Mamdani engine with COG method for the defuzzification of the outputs. To verify the validity of the results we have implemented a simulator described after in the paper.

4 Neuro fuzzy controller

Another solution we have realized is based on adaptive neuro fuzzy networks (ANFIS).

There are two neuro fuzzy networks, everyone with two inputs and one outputs. To train the networks is been made up a mathematic model based on the observation 1) and 2) of the paragraph “Problem formulation”. Based on them emerge that for increasing of angle \( \theta l \) and distance \( RO \), the angle \( \theta_{out} \) decrease so there is an inverse proporzionality. So we have built two function called \( \rho \theta \) and \( \rho d \) that weighs respectively the input for the angle and the input for the distance to obtain a formulation for the output \( \theta_{out} \) so to consider the proporzionality in advance:
\[ \theta_{\text{out}} = -\text{sgn}(\theta l) \times 90 \times \rho d \times \rho \theta \]

\[ 0 \leq \rho d \leq 1 \quad 0 \leq \rho \theta \leq 1 \]

The minus sign is due to the convention for the angle \( \theta l \). The form of the function \( \rho \theta \) and \( \rho d \) is a quadratic curve, that go through typical point we have established, made up with Curve Fitting Toolbox of Matlab:

\[ \rho_0 = 7.778 \times 10^{-0.05} \times 0.1^2 - 0.1811 \times 0 + 1 \]

\[ \rho_d = 8.333 \times 10^{-0.05} \times \rho^2 - 0.1833 \times 1 + 1 \]

If \( \theta l = 0 \) the obstacle is along the direction between the robot and the target so the robot steering angle have to be maximum (90 or -90 degree) that is the weigh to apply have to be one. In fact when \( \theta l = 0 \) the function \( \rho \theta \) is one. In this situation it’s necessary consider the distance between the robot and the obstacle: if the obstacle is near then the weigh \( \rho d \) has to be big so \( \rho d \times \rho \theta \) will be one, instead if the obstacle is far from the robot then \( \rho d \) has to be little so \( \rho d \times \rho \theta \) will be approximately zero.

For the movement we consider that the robot detect always nearest obstacle. Hence a safe movement can be:

\[ \text{Movement} = \text{RO} \times \cos(\theta l) \]

4.1 Training set generation

Adaptive neuro fuzzy network required a prior knowledge (training set) for modelling the workspace. For the design of the ANFIS we have used the ANFISedit Toolbox of Matlab. A restriction of this toolbox is that through it it’s possible design only one output networks. This cause our choose to create two networks.

The training set is the same for both networks and it has been generated based on mathematic model described in the previous paragraph. We have generated the values for the angle \( \theta l \), included in the range -90° e +90°, and the values for the distance RO, in the range 0 ÷ 100, randomly. The outputs \( \theta_{\text{out}} \) and Movement have been calculated with the mathematic model.
With ANFIS the inferential engine is Sugeno type. The number of training epochs has been fixed to 250 and we have obtained an error for the first network equal to 2.7451 degree and an errore for the second network equal to 0.14987.

After several test it has been obtained a good generalization for new input – output couple unknown for the networks.

5 Optimisation of the fuzzy controller
One of the central factors in the design of efficient and robust fuzzy logic controllers is the selection of the membership rules and parameters. The existing approaches for choosing the membership functions are based on trial-and-error process, and lack learning and autonomy[8]. One method of removing the uncertainty associated with the selection of these variables is the use of genetic algorithms (GA). These have been widely used in a variety of problems especially in machine learning and function optimisation. GAs were created by John Holland[9] in the 1960’s13 and proved capable of solving complex problems in large scale of applications. Genetic algorithms do not have any special requirements regarding the fitness function, thus granting maximum liberty in choosing an appropriate function[10]. All they need are a lot of genes, which represent encoded parameter sets. The totality of the genes represent the chromosome that is a potential solution of the optimisation problem; in this contest the fitness function describes the quality of the solution. In our case it’s possible to consider the fitness function like the error function turn in sign based on difference between the outputs of inferential engine and the correspondet outputs of the mathematic model used for the neuro fuzzy networks. The requirement to consider the error changed in sign is originate from the fact that usually in the genetic algorithm theory is maximized the fitness function; we want to minimize the error so we maximize the error changed in sign.

Membership function used in fuzzy system are triangular and trapezoidal. So they present three or four parameters:

The chromosome for our problem is:

| a1 | b1 | c1 | ... | ... | a2 | b2 | c2 | d2 | ... | ... | ... |

Fig. 18 Encoded parameter set

Fig. 19 Chromosome

where a1, b1, ecc are the encoded parameters in the single gene.
The domain for the distance is 0 ÷ 100; that one for the angle is -90 ÷ 90 degree.
For the genes of the distance domain, it’s necessary to have enough bit to encode at least the value 100; so we can use 7 bit for every gene that encoded the parameters for the membership function in the distance domain. With 7 bit it’s possible obtain up to 128 configurations of bit; so we have to discart those chromosome that presents inadmissible configurations.
At the same way, for the genes of the angle domain we can
use 8 bit to encode at least the value 180 (we shift the
range -90 ÷ 90 into 0 ÷ 180).
The chromosome generation occurs by applying the
evolution operation that are:

- Crossover: creates a new chromosome by
  combining two existing chromosomes with high
  fitness ratings and mutation occasionally randomly
  change the value of a gene in a chromosome;
- Mutation: occasionally randomly change the value
  of a gene in a chromosome, a 0 is changed to a 1
  and vice versa; usually this operation is made up
  after the crossover

In our case after every evolution, the value of the genes of
every chromosome have to be take back in the
respondent parameter in a MF and with the totality of
the MF defined in this way it’s necessary to determine the
fuzzy outputs correspondent to a set of inputs (a prior
fixed, like a training set) and compare them with the
outputs correspondent to the same set of input obtained
with mathematic model used for the neuro fuzzy networks.
For every comparison we have an error so the totality of
the errors constitutes an error vector, so we can consider
like fitness function the medium quadratic error changed in
sign that ponders all the component of the vector.

In Internet it’s possible to find a large number of
freeware software that allows to manipulate genetic algorithm
simply and specially allows to lose one of the evolution operations demands only the definition of the
chromosome’s structure (in term of the type of the genes,
ad example Boolean, integer, ecc), the definition of the
fitness function and the number of the individual that will
constitute the initial population.

In the follow it has been analyzed three typical situation:
- obstacle is along the direction between robot and
target;
- obstacle is on the right side of the direction
  between robot and target;
- obstacle is on the left side of the direction between
  robot and target;

7 Conclusions
In this paper the problem about navigation of mobile
robot in unknown environment is solved using more
popular soft computing methodology, that is fuzzy logic
and neuro fuzzy, locating in the genetic algorithms a
possible optimisation solution. From the results obtained
it is possible to conclude that the design answer to the
prefixd targets regarding the tollerances and the made work supposition.

References:
Senia, “Fuzzy Logic Controller for Behaviour-
Based Navigation Systems”, WSEAS
TRANSACTIONS on CIRCUITS and SYSTEMS,

Schema for Mobile Robots”, WSEAS
TRANSACTIONS on SYSTEMS, pp. 2211-2214,
July 2004.

Mobile Robots in Unknown Enviroments”,
Journal of Intelligent and Robotic Systems 37,
2003.

Avoidance for fast Mobile Robot,” IEEE
Transactions on System Man and cybernetics,

Mobile Robot Using Fuzzy Logic and
Reinforcement Learning,” IEEE Transactions on
Systems, Man, and Cybernetics, Vol. 25, No. 3,
(March 1995).

Logic Based Algorithm for a Mobile Robot Collision
Avoidance in an Unknown Environment.

alle reti caotiche e neuro-fuzzy”, Etas Libri

Integration Of An Optimised Fuzzy Logic Navigation
Algorithm Into A Semi-Autonomous Robot Control
System”


[10] Reinhard Braunstingl, Jokin Mujika, Juan Pedro U
ribe, “A Wall Following Robot With A Fuzzy Logic
Controller Optimised By A Genetic Algorithm”