A NEW CONTROLLER FOR MOBILE ROBOTS BASED ON FUZZY LOGIC AND GENETIC ALGORITHMS

SALVATORE PENNACCHIO, TOMMASO RAIMONDI, FRANCESCO MARIA RAIMONDI, GIUSEPPE BOSCO
Dipartimento di Ingegneria dell'Automazione e dei Sistemi
University of Palermo
Viale delle Scienze
ITALY

Abstract: - This paper shows a new model of controller for navigation of mobile robots in static environments with fixed obstacles. In particular, our controller is based on fuzzy logic. The learning process, that is indispensable for the improvement of the robotic motion and for the reaching of the final goal in a fluent way, is realized by using a new technique of soft computing, based on modern theories concerning genetic algorithms.

Key-Words: - Mobile Robot, Fuzzy Logic, Genetic Algorithm

1 Introduction
Mobile robots play a very important role in flexible manufacturing systems. In addition they are applied in construction automations, military missions, interplanetary explorations, and so on. The main issues in mobile robots research include mission planning, navigation, maneuvering and manipulation [1]. The design of an autonomous robot is a complex task and the criteria of success are evaluated in terms of its capabilities to make decisions and to act by itself in a reliable and satisfactory manner [2]. In the last years, autonomous mobile robots are required to navigate in more domains, where the environment is uncertain and dynamic. Autonomous navigation in these environments demands adaptation and perception capabilities [3]. In order to navigate in unknown environments, a mobile robot needs to deals with the environments in a timely manner. This results in real-time demands on the navigation system. Due to its simplicity and capability for real-time implementation, fuzzy logic is an excellent candidate for such applications[4]. Fuzzy navigation systems are simpler to implement than other navigation systems because they can handle infinite navigation situation with a finite set of rules[5]. In the context of mobile robot control, a fuzzy logic based system has the advantage that it allows the intuitive nature of sensor-based navigation to be easily modeled using linguistic terminology. As a result, reactive fuzzy control systems permit intelligent decisions to be made in real time, thus allowing for smooth and uninterrupted motion [6]. Autonomous navigation of a mobile robot has been studied by many researches [7, 8, 9, 10],This paper presents a new controller for navigation of mobile robots in static environments. In particular, our controller is based on fuzzy logic while the learning process, that is indispensable for the improvement of the robotic motion, is realized by using a new technique of soft computing, based on modern theories concerning genetic algorithms.

This paper is organized as follows. Section 2 is for related work. Section 3 shows the planning of a fuzzy controller and its rules. Section 4 shows the fuzzy controller new method of learning, based on genetic algorithms. Finally section 5 shows our conclusions.

2 Related work
For navigating mobile robots, numerous approaches have been developed in recent years. Several fuzzy strategies have been applied to this problem obtaining in most cases satisfactory results. A fuzzy controller used to control an obstacle avoidance mobile robot is presented in [11]. This controller uses three sub-controllers. The outputs are summed to produce a concerted effort to control the robot away from obstacles. A fuzzy navigation system that can escape from concave and maze-like obstacle fields in an unknown environments is presented in [12]. The system combines a tangent algorithm for planning with sets of linguistic fuzzy control rules. In [13] a real mobile robot with a fuzzy controller has been implemented and tested in

![Fig. 1 Fuzzy controller structure](image-url)
various environments. In [14] a new technique to obtain a fuzzy perception of the environment, dealing with the uncertainties, imprecisions and blind sectors from the sensorial system has been introduced. This fuzzy perception is used to implement a fuzzy behavior-based control architecture. In [15] is presented a fuzzy logic controller for mobile robot navigation. The designed fuzzy controller maps the input space to a safe collision-avoidance trajectory in real time. The technique generates satisfactory direction and velocity maneuvers of the autonomous vehicle which are used by the robot to reach its goal safely. Fuzzy logic can be utilized for the fusion of multiple neural networks. In [16] a neural integrated fuzzy controller which integrates the fuzzy logic representation of human knowledge with the learning capability of neural networks is presented. This architecture comprises: 1) Fuzzy logic membership functions, 2) a rule neural network, 3) an output-refinement neural network. In [17] is presented a neuro-fuzzy system for localising mobile robots solely based on raw vision data without relying on landmarks or artificial symbols. A four-step hybrid method for the design of neuro-fuzzy motion controllers is presented in [18]. The proposed method uses in an efficient way various technique of the computational intelligence (fuzzy systems, neural networks and genetic algorithms). Finally a neuro-fuzzy approach in order to guide a mobile robot is presented in [19]. The approach proposed is able to extract a set of fuzzy rules set from a set of trajectories provides by a human. These trajectories guide the mobile robot towards the target in different cases and it is possible to obtain the rules and membership functions automatically.

3 Control system and fuzzy rules
The fuzzy controller has been built in order to allow the robot to improve its motion and to reach the final goal. In particular, when the robot meets an obstacle during its shiftings, the fuzzy controller enters upon office which allow the robot to avoid the obstacle in a linear and fluent way.

Figure 1 shows in detail the structure of the realized controller.

The parameters used as inputs to the fuzzy controller are information coming from environmental sensors, that is from those devices of the mobile robot and they enable it to reconstruct the surrounding environment. In particular as inputs of the controller we used the following variables:

- \( \text{distance}\_\text{obstacle}_\Delta x \): indicates the distance along the horizontal axis between the mobile robot and the obstacle;
- \( \text{distance}\_\text{obstacle}_\Delta y \): indicates the distance along the vertical axis between the mobile robot and the obstacle;

Both \( \text{distance}\_\text{obstacle}_\Delta x \) e \( \text{distance}\_\text{obstacle}_\Delta y \) are considered in comparison with a fixed obstacle, as shown in figure 2.

As far as the fuzzy controller outputs are concerned, we used the following variables:

- \( \text{motor}\_\text{robot}_x \): indicates the direction along the horizontal axis that the mobile robot in order to avoid the obstacle;
- \( \text{motor}\_\text{robot}_y \): indicates the direction along the vertical axis that the mobile robot in order to avoid the obstacle;

We considered two possible situations in which the robot can be situated: in the first one, we suppose that the robot moves to right and/or to the top, while the second situation shows how the robot moves to right and/or to the bottom. If we consider the first situation, we used three linguistic fuzzy sets \{NEAR, MEDIUM, FAR\} for \( \text{distance}\_\text{obstacle}_\Delta x \) with its membership functions of trapezoidal and triangular shape as shown in figure 3.

![Fig. 2 Distance between the obstacle and the mobile robot](image)

![Fig. 3 Membership functions in distance_obstacle_\Delta x](image)

![Fig. 4 Membership functions in distance_obstacle_\Delta y](image)
The same thing as far as distance_\text{obstacle}_y is concerned, as shown in figure 4.

As far as the fuzzy controller outputs are concerned we used these linguistic sets \{FORWARD, HIGH FORWARD\} in relationship with the motor_\text{robot}_x variable. Whereas the linguistic sets \{LEFT, RIGHT\} have been used with the motor_\text{robot}_y variable.

The respective membership functions are shown in figure 5 and 6.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig5.png}
\caption{Membership functions in motor_\text{robot}_x}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig6.png}
\caption{Membership functions in motor_\text{robot}_y}
\end{figure}

We used the Mandami model: the defuzzificated results represent the components along the horizontal and vertical axis which will define the motion of the mobile robot in order to avoid the obstacle it meets. We used operators or/and, as shown in the following formulas:

\begin{align}
\mu_{\text{or}}(x) &= \max(\mu_a(x), \mu_b(x)) \\
\mu_{\text{and}}(x) &= \min(\mu_a(x), \mu_b(x))
\end{align}

The outputs have been calculates using the following formula:

\begin{equation}
\text{output}_{\text{ crisp}} = \frac{\sum_i \mu_i(\text{output}_i)}{\sum_i \mu_i(\text{output}_i)}
\end{equation}

The fuzzy rules we used to get the output variables are summarized in the following table, which refers to the robot motion towards right and/or the top:

\begin{table}[h]
\centering
\begin{tabular}{ |c|c|c|c| }
\hline
\text{Distance_\text{obstacle}_x} & \text{NEAR} & \text{MEDIUM} & \text{FAR} \\
\hline
\text{NEAR} & \text{LEFT} & \text{FORWARD} & \text{HIGH} \\
\hline
\text{MEDIUM} & \text{LEFT} & \text{FORWARD} & \text{HIGH} \\
\hline
\text{FAR} & \text{FORWARD} & \text{HIGH} & \text{FORWARD} \\
\hline
\end{tabular}
\caption{Fuzzy rules y>0}
\end{table}

In table 2 we report the fuzzy rules used when the mobile robot moves towards right and/or towards the bottom, that is when the component along the vertical axis of \(y\) is negative:

\begin{table}[h]
\centering
\begin{tabular}{ |c|c|c|c| }
\hline
\text{Distance_\text{obstacle}_y} & \text{NEAR} & \text{MEDIUM} & \text{FAR} \\
\hline
\text{NEAR} & \text{RIGHT} & \text{RIGHT} & \text{FORWARD} \\
\hline
\text{MEDIUM} & \text{RIGHT} & \text{FORWARD} & \text{HIGH} \\
\hline
\text{FAR} & \text{FORWARD} & \text{HIGH} & \text{FORWARD} \\
\hline
\end{tabular}
\caption{Fuzzy rules y<0}
\end{table}

\section{Learning and Training}

The fuzzy controller has been trained by using a genetic algorithm. The algorithm we used for the learning process of the fuzzy controller is the basic genetic algorithm, consisting in the following points:

1. Casual initialization of a population of \(M\) chromosomes and calculation of the fitness function;
2. Casual choosing of two chromosomes and the execution of the genetic cross-over operation, obtaining two new chromosomes that will become its respective parents and the calculation of the corresponding fitness function;
3. Casual choosing of a string and its bit and the execution of the mutation operation, obtaining a new chromosome of which we can estimate the fitness function;
4. Reproduction of \(M\) chromosomes;
5. Back to point 2 to the reaching of a prefixed criterion;
Figure 7 shows in detail the structure of the control system proposed.

In particular we considered the defuzzificated controller outputs, which represent the directions along the horizontal and vertical axis that the mobile robot must follow in order to avoid the obstacle it meets during its wandering. These outputs depend on the values allotted to the input variables of the controller. This outputs also depend on the values allotted to the corresponding membership functions. Then we calculated the error $E$ shown in the following formula:

\[ E = Y - D \]

in which $Y$ represents the output of the fuzzy controller and $D$ represents the value desired.

The goal we wanted to reach was that vector $E \to 0$ as much as possible. On the contrary we applied the genetic algorithm. A population of $M$ chromosomes was initialized and the fitness function for each of those was then calculated. As far as the fitness function is concerned, this was chosen by keeping accounts of the Euclid distance between the mobile robot and the fixed obstacle and by trying to minimize the distance between them:

Then we effected the genetic operation of cross-over and mutation and so we determined new possible solutions generations and we applied repeatedly the genetic algorithm in order to obtain a $E \to 0$ error.

After the design phase the fuzzy controller was implemented and then we made several experimental tests that showed effectiveness of the proposed fuzzy controller.

5 Conclusions

In this paper we proposed a new controller for the navigation of mobile robots in static environments. The controller we realized is based on the fuzzy logic while the learning process, necessary in order to improve the robot motions and to allow it to reach the final goal in a smooth and fast way, was realized by using a new soft computing technique based on the most recent theories on the genetic algorithms. The control system we proposed in this paper will be continued.

In the future the control system can be improved and will be possible to add new membership functions and characteristic parameters can be considered for the accomplishment of more complicated and intelligent strategies.

6 References


