BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LX No. 3B/2008	67 - 72	Seria Tehnică

Implementation of Neuro - Fuzzy Controller on Khepera III Mobile Robot

Cristina Popescu, Otilia Cangea

Universitatea Petrol – Gaze din Ploieşti, B-dul Bucureşti nr.39, Ploieşti
e-mail: cristinap@upg-ploiesti.ro

Abstract

Avoiding obstacles in navigation of mobile robots in unstructured environments is an actual problem for scientists, but for mostly; the results are obtained by simulation. In this paper, a neuro-fuzzy based approach is proposed, which coordinates the sensors information and robot motion together. A fuzzy logic system is designed for obstacle avoidance behaviour and it is used a Takagi-Sugeno controller which learn like a neural network, so we can call a neuro-fuzzy controller. The efficiency of the proposed approach is demonstrated by simulation studies in Matlab language and by experiments for real implementation on the Khepera III mobile robot.

Key words: *mobile robot, neural network, fuzzy, neuro-fuzzy control.*

Introduction

A mobile robot is a non-linear plant which is difficult to model. Hence, it is a good candidate as an experimental platform for the validation of fuzzy and neuro-fuzzy controllers. Furthermore, the state variables of a mobile robot are easy to visualize because they have a intuitive relation to the robot behaviour. Therefore, the linguistic if-then rules could be define in an intuitive way. The problem occurs when a robot has many sensors and actuators. The complexity of the controller increases and the construction of the rule base is more complicated, especially if a complex behaviour is required.

In this paper we will describe experiments with the Khepera III mobile robot, developed in 2007 in the Microcomputing Laboratory at the Swiss Federal Institute of technology. The earliest versions of the Khepera robot were available since 1992.

This problem was similarly studied for Khepera II robot [1], but for this robot the sensorial system was change, so we adapted the control structure. We also developed a Matlab functions to communicate with the robot, via bluetooth connection, using a Windows platform.

Robot Configuration

Our experiments are made with a robot named Khepera III. It is cylindrical in shape, measuring 130 mm in diameter and 70 mm in height and its weight is 690 g. It is shown in figure 1.

Fig. 1. Khepera III mobile robot

The basic configuration of Khepera is composed of DsPIC 30F5011 at 60 MHz processor, system and user memory, extension busses and a serial link. The microcontroller includes all the features needed for interfacing with memories, with I/O ports and with external interruptions.

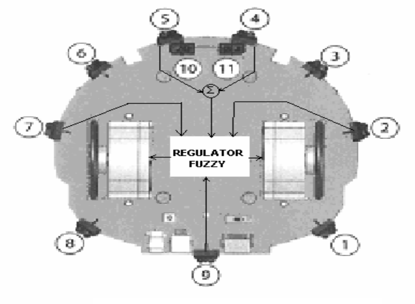
The sensory/motor board includes two DC motors with incremental encoders, 9 infra-red proximity and ambient light sensors with up to 25 cm range, 2 infra-red ground proximity sensors for line following applications and 5 ultrasonic sensors with range 20 cm to 4 meters.

Neuro-Fuzzy Controller for Implement Obstacle Avoidance Behaviour

In this paper we present controller design for obstacle avoidance of the Khepera III with a minimal number of linguistic rules. The robot has 9 infra-red proximity sensors and 2 motors. The inputs are the linguistic variables: distances *between the robot and the obstacle* and the outputs are linguistic variables: *motor speeds*. We work with four input linguistic variables: *distance to the left* D_s, *distance to the front* D_f, *distance to the right* D_d and *distance on the back* D_{sp}, like are shown in figure 2.

Distance to the front D_f

Distance to the left D_s



Distance to the right D_d

Distance on the back D_{sp}

Fig. 2. Inputs variables of Khepera III

If we note with S_1, ..., S_{11} the sensor values normalised within [0,1], then the input variables are calculated in the following way:

$$D_s = S_7 \tag{1}$$

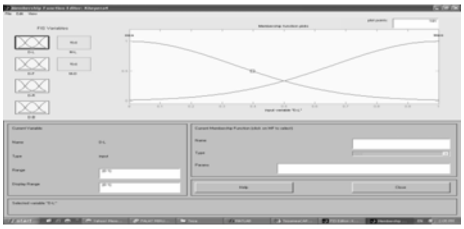
$$D_f = \frac{S_4 + S_5}{2} \tag{2}$$

$$D_d = S_2 \tag{3}$$

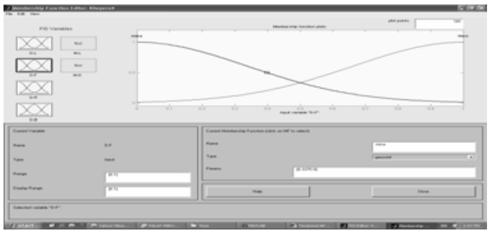
$$D_{sp} = S_9 \tag{4}$$

For each input linguistic variable, we define 3 linguistic values: *small, medium, big*, and for each output we define 7 linguistic values: *backward fast, backward medium, backward slow, stop, forward slow, forward medium, forward fast* [3].

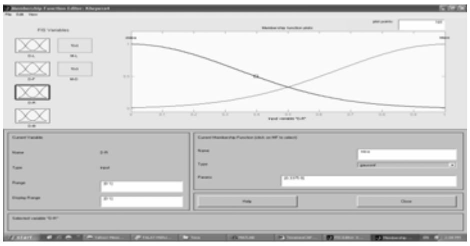
The first step to design the controller is up to set the initial base of the rules and the parameters of the controller. We set 16 rules, which constitute the initial base rules, to implement the obstacle avoidance behaviour. The second step is to implement the controller. The Takagi Sugeno controller's parameters will be adapt by a supervised learning method based on gradient descent, which consists of modifying parameters in order to obtain the desired output in response to given inputs. A supervised learning method is an identification algorithm. We choose an on-line identification method because it uses less computational time and less memory space than an off-line algorithm. After learning, linguistic rules are extracted from the system parameters. They may be different than the initial rules. The parameters of membership functions are initialized to uniformly cover the input space. Each membership function for input variables are Gaussian functions and for output variables are singleton functions, like we shown in next figures.



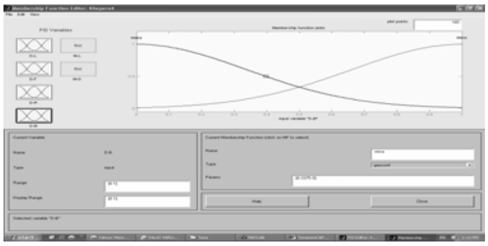
a) Distance to the left D_s



b) Distance to the front D_f



c) Distance to the right D_d



d) Distance on the back D_{sp}

Fig. 3. Membership functions for input variables.

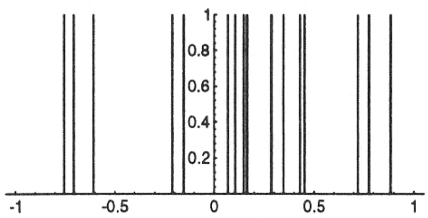


Fig. 4. Membership functions for output variables

The initial base rules describing obstacle avoidance behaviour contain 16 rules and is presented in table 1. We was demonstrated that with 16 rules, the Khepera avoids obstacle with success, but adding rules can create an inconsistent rule base or make conflicts between the rules.

The particularity of the Takagi and Sugeno's controller is that the consequent parts of linguistic rules are expressed as functions of linguistic variables, and in this case we will consider only the special type of Takagi and Sugeno's controller were these functions are constants [4].

Table 1. The initial base rules of the controller

Rule	**D_s**	**D_f**	**D_d**	**D_{sp}**	**v_s**	**v_d**
R1	Small	Small	Small	Small	Stop	Stop
R2	Small	Small	Small	Big	Backward fast	Backward slow
R3	Small	Small	Big	Small	Backward slow	Forward fast
R4	Small	Small	Big	Big	Forward medium	Backward medium
R5	Small	Big	Small	Small	Backward fast	Backward fast
R6	Small	Big	Small	Big	Forward medium	Stop
R7	Small	Big	Big	Small	Backward slow	Stop
R8	Small	Big	Big	Big	Forward medium	Forward medium
R9	Big	Small	Small	Small	Backward fast	Forward fast
R10	Big	Small	Small	Big	Forward medium	Forward fast
R11	Big	Small	Big	Small	Forward slow	Backward medium
R12	Big	Small	Big	Big	Backward slow	Backward slow
R13	Big	Big	Small	Small	Backward slow	Forward medium
R14	Big	Big	Small	Big	Backward medium	Backward slow
R15	Big	Big	Big	Small	Stop	Backward slow
R16	Big	Big	Big	Big	Backward medium	Forward medium

The Takagi and Sugeno's fuzzy controller has three types of parameters to adapt:

- centre values $a = (a_{11}, \ldots, a_{nm}, \ldots, a_{NM})^T$,
- width values $b = (b_{11}, \ldots, b_{nm}, \ldots, b_{NM})^T$,
- consequent values $c = (c_{11}, \ldots, c_{nk}, \ldots, c_{NK})^T$.

For Gaussian membership functions of the controller, the adaptation of the parameters and weights is done by:

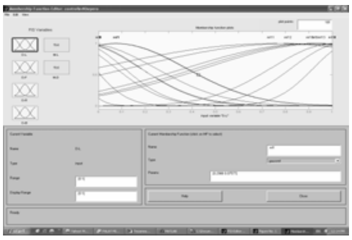
$$a_{nm}(t+1) = a_{nm}(t) - \Gamma_a \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{x_m(t) - a_{nm}(t)}{b_{nm}(t)^2} \sum_{k=1}^{K} \left(y_k(t) - y_{d_k}(t)\right)\left(c_{nk}(t) - y_k(t)\right) \tag{5}$$

$$b_{nm}(t+1) = b_{nm}(t) - \Gamma_b \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{\left(x_m(t) - a_{nm}(t)\right)^2}{b_{nm}^3} \sum_{k=1}^{K} \left(y_k(t) - y_{d_k}(t)\right)\left(c_{nk}(t) - y_k(t)\right) \tag{6}$$

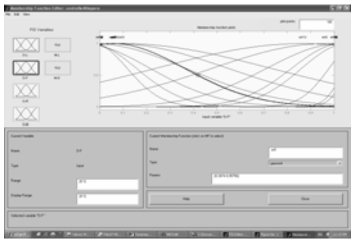
$$c_{nk}(t+1) = c_{nk}(t) - \Gamma_c \frac{u_n}{\sum_{n=1}^{N} u_n} \left(y_k(t) - y_{d_k}(t)\right) \tag{7}$$

During the learning process the rules base change and will observe that there are no contradictions. It means that are not two rules with the same antecedents parts and different consequent parts. We used the Min-Max method for the inference process and Centre of Gravity method for defuzzification.

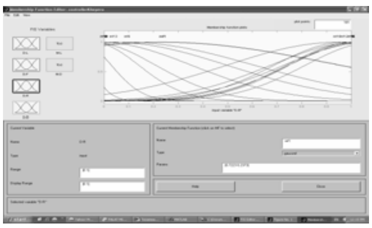
After learning process the membership functions are modified like in following figures:



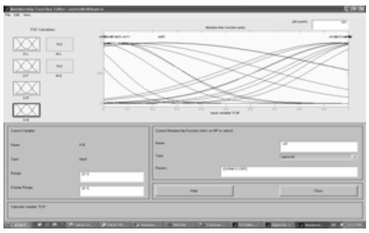
a) Distance to the left D_s



b) Distance to the front D_f



c) Distance to the right D_d



d) Distance on the back D_{sp}

Fig. 5. Membership functions after learning process

It is observed that after 500 iterations in the learning process, Khepera learned the desired behaviour and the value of error is good, E=0,0198, the evolution of error being like in next figure.

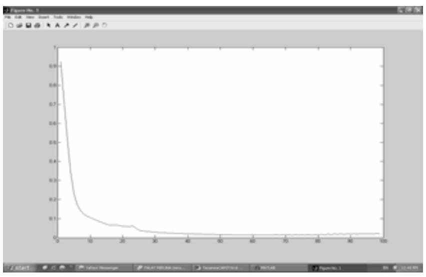


Fig. 6. Evolution of error in learning process.

When implementing the Takagi and Sugeno's controller in the real robot Khepera III, we observe that with 16 rules the robot avoid obstacles with success and by changing the rules base, we can implement another behavior, like wall following or reached a target.

Conclusion

In this paper we studied the design and the implementation of Takagi and Sugeno's controller, named neuro-fuzzy controller. For the first time, this problem was studied by Godjevac (1997) [1],[2], but we adapt the method and the algorithm for the new Khepera III. The neuro-fuzzy controller was first validated in simulation, then applied to the control of a real mobile robot.

The first step is to build an initial set of linguistic rules, which is set up to run its assigned task. For adjust the parameters of the fuzzy controller we used an adaptive algorithm similar to a neural network learning. The learned knowledge can be represented in the form of linguistic rules, which can be extracted from the adapted parameters of the fuzzy controller and this is considered as the biggest advantage of a neuro-fuzzy network compared to a neural network.

References

1. Godjevac, J. – *Neuro-Fuzzy Controllers Design and Application*, Presses Polytechniques et Universitaires Romandes, ISBN 2-88074-355-9, Lausanne, 1997
2. Godjevac, J. , Steele, N. – *Neuro-Fuzy Control for Basic Mobile Robot Behaviours*, In : Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Driankov, D., Saffiotti, A., pp. 97-117, Physica-Verlag Heidelberg, 2001, ISBN 1434-9922
3. Popescu C. – *Mobile Robot Navigation Using Fuzzy Logic For Obstacle Avoidance*, Bulletin of "Petroleum-Gas" University, 5th International Symposium on Process Control SPC 2006, Ploiesti, ISSN 1224-8495.
4. Sugeno, M., Murakami, K. - *An Experimental Study on Fuzzy Parking Control Using a Model Car*, In: Industrial Applications of Fuzzy Control, Sugeno M. (Ed.), pp. 125-138, North-Holland, 1985.

Implementarea regulatorului neuro-fuzzy pe robotul mobil Khepera III

Rezumat

În cadrul acestei lucrări am abordat o problemă de actualitate pentru cercetători, şi anume, evitarea obstacolelor în procesul de navigare a roboților mobili într-un mediu necunoscut. În general, toate rezultatele care vizează această problemă au fost obținute prin simulare. In acest articol am propus o metodă de conducere neuro-fuzzy prin care sunt coordonate mişcările robotului cu informațiile primite de la senzori. Pentru aceasta am utilizat un regulator Takagi şi Sugeno care are proprietatea de a învăța ca o rețea neuronală, deci poate fi numit regulator neuro-fuzzy. Eficiența metodei propuse este demonstrată prin simulări în limbajul de programare Matlab şi prin experimente pe robotul mobil Khepera III.