BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi" din Iaşi,
Tomul LVI (LX), Fasc. 4, 2010
Secţia
CONSTRUCŢII DE MAŞINI

AUTONOMOUS ROBOT NAVIGATION BASED ON FUZZY LOGIC AND REINFORCEMENT LEARNING

BY

CRISTINA GABRIELA ROZMARIN*, SID AHMED BERRABAH**

Abstract: This paper presents a behaviour-based algorithm for autonomous mobile robot navigation in an unknown environment. The multi - behaviour is represented by the coordination between tasks like reaching a certain point and avoidance of any collisions, based on the information received from a group of sensors. The method used to achieve this goal consists of a fusion between two strategies: fuzzy logic and unsupervised learning approach – reinforcement learning, which allows the system to evolve and improve its performances. The navigation process is adapted for the ROBUDEM robot, implemented as CoRoBA module and tested using the CoRoSim simulation platform.

Key words: robot navigation, fuzzy logic, reinforcement learning, fuzzy controller

1. Introduction

The theory of fuzzy logic was first presented by Lotfi Zadeh and was welcomed with scepticism by the scientific community, but turned out to be of a great success concerning the process of producing new technologies used in implementing complex and sophisticated systems. Fuzzy Logic makes a reliable method suited for the domain of Robotics, as it allows working with ambiguous variables, as these examples show [2], [4] and [6].

This paper presents a multi-behavior strategy for autonomous robot navigation in an unknown environment. The multi-behavior is represented by the coordination between tow tasks: one for reaching a target point and the other for obstacle avoidance based on the data received from a set of sensors[7]. The obstacle avoidance behavior is based on an adaptive fuzzy logic controller using the reinforcement learning algorithm for itself improvement. The navigation process is adapted for the ROBUDEM robot, implemented as CoRoBA module and tested using the CoRoSim simulation platform.

2. Fuzzy logic and control

In the last decades, fuzzy logic became an important developing tool for applications designed to work on systems which are affected by uncertainty, imprecision or vagueness, or for which an accurate mathematical model cannot be easily obtained[3].

Fuzzy logic extends the values of the degree of membership to an interval, which contains numbers between 0 and 1. Mathematically, a fuzzy set A over a universal set X is defined by its membership function:

(1) $$\mu_A(x) : X \rightarrow [0,1]$$

Using the fuzzy sets and variables and the fuzzy operators the conditional rules can be created. The syntax of these rules is:

IF x is A THEN y is B,

where x and y are fuzzy variables, and A and B are fuzzy sets.

2.1 Fuzzy Controllers

Fuzzy controllers are used to describe the states of a system and the transitions between them. The main parts of a fuzzy controller are: Fuzzifier, Fuzzy rule base, Fuzzy inference engine and Defuzzifier[1].

The “Fuzzifier” component is responsible for the fuzzification of the input variables, which are transformed into linguistic variables, with the help of the membership functions.

After the process of fuzzification, using the “Fuzzy rule base”, the input data is interpreted and processed and the result is obtaining a unique solution.

The most common method to combine into one unique conclusion the different results obtained from processing the rules and the input data, is the MAX-MIN Inference. Therefore the output is a fuzzy set whose membership function is of the form:

(2) $$\mu_y = \max_i(\min(\mu_{x_1^i}(x_1)...\mu_{x_n^i}(x_n)), \mu_{y^i}(y))$$

To obtain a crisp value which can be read as a command measure, the output must pass through the “Defuzzification” component. The most commonly used method is the Centroid Technique:

(3) $$x = \frac{\sum_i x_i \cdot \mu_A(x_i)}{\sum_i \mu_A(x_i)}$$

where x_i belongs to the universal set A with a grade of membership of $\mu_A(x_i)$.

2.2 Reinforcement learning for fuzzy controllers

For designing a complex and accurate system, the artificial intelligence alone is not sufficient. There must be used a learning approach as well. Reinforcement learning is a part of the unsupervised learning domain, which means learning without a teacher[5]. The highlight of this method is that the agent is capable of operating real-time changes in the environment.

In order to use the Reinforcement Learning approach, the Q-learning algorithm was chosen. The main characteristic of this algorithm is: one can associate for a rule both the action and also the quality of the action (q value).

For the learning of the fuzzy controller, several possible actions are affected to each rule of the controller and the learning process consists of choosing the action with the best quality.

The Q-learning algorithm is explained below[5]:

1. Initialize Q(s, a) to 0 for all state *s* and action *a*
2. Perceive current state *s*
3. Choose an action *a* according to action value function
4. Carry out action *a* in the environment. Let the next state be *s'* and the reward be *r*.
5. Update action value function from *s, a, s'*, and *r*:

$Q_{t+1}(s, a) = Q_t(s, a) + \alpha\, [\, r_{t+1} + \gamma \max Q_t(s', a') - Q_t(s, a)\,],\ 0 < \alpha < 1$

, where α is a learning rate parameter and γ is a fixed discounting factor between 0 and 1.

6. Return to 2

3. Behavior based robot navigation

The purpose of this work is to develop an application in which a robot must find its path and avoid any collision with obstacles. In order to achieve this goal, fuzzy logic controllers are used to build two basic behaviours: Goal Seeking and Obstacle Avoidance. Once the behaviours have been identified, the coordination between them must be defined: if no obstacle is detected in the surrounding of the robot, the goal seeking controller is activated, otherwise the obstacle avoidance behaviour decides for the robot how to avoid the detected obstacle.

3.1 Goal Seeking Controller

The Fuzzy controllers receive as inputs processed data from the sensors and output commands for the actuators.

The Goal Seeker Controller is responsible for the good navigation of the robot and its orientation towards the point it must reach.

This Fuzzy Controller is a "Sugeno" type and works similar with the human way of thinking: if the goal is on the right and the distance is big then make a small turn to the right with big velocity. If the goal is on the right and the distance is small then make a big turn to the right with small velocity.

From Figure 1 it can be observed that the goal direction is more important than the goal distance. The output angle increases with the goal angle. If the distance is very short, the turn angle increases rapidly and if the distance is long, the dependency on the goal direction is less important.

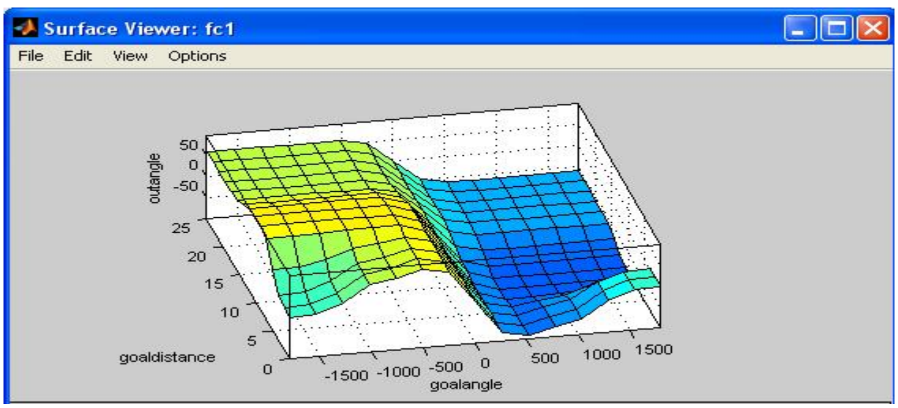

Fig. 1 - The turn angle and the velocity as functions of goal distance and goal direction

3.2 Obstacle avoidance controller

Given the data received from the sonar sensors, the Obstacle Avoidance Controller calculates the turning angle of the robot in order to avoid obstacles. This fuzzy logic controller is also a "Sugeno" type.

The robot is equipped with four pairs of sonar sensors displayed on all sides of the robot. When an obstacle is considered close to the robot, the Goal Seeking behavior is deactivated, and the Reinforcement learning algorithm for the Obstacle Avoidance is activated. Once this being accomplished, the membership degree for each input is calculated in order to choose the active rule from the rule base. Example of a rule (rule number 1):

If x_1 is A_1 and x_2 is A_2 … then $y = a[1, 1]$ with $q[1, 1]$
$y = a[1, 2]$ with $q[1, 2]$
$y = a[1, 3]$ with $q[1, 3]$

, where y represents the output of the next action – in this case the turning angle was considered and q represents the associated quality value of the chosen action.

After the rule is activated, an output of the three must be chosen. The consequences of applying an action are evaluated by the algorithm. If the next state is a convenient one, the agent receives a reward (the q-value will be increased proportionally with the value of the reward) and will continue to choose the same action, as it will have the maximum q value. Otherwise, will receive a punishment (the q value will be decreased based on the value of the punishment) and the action with the maximum q value will be chosen. In the next steps of the algorithm all the q values are updated based on the previous taken action. When restarting the algorithm the agent will know exactly which the best action to apply is.

4. Implementation and results

To be able to test the application, a simulation environment is needed, the MoRoS3D. The virtual robot is capable of moving in a 2D world populated with other robots or obstacles. The experiments which have been done in order to test the two fuzzy controllers and the implementation of the reinforcement learning algorithm point out that the robot reaches the received goals following a free of collisions path.

As Figure 2 shows, there is an obstacle in the robot's path, which is detected by sensors. In order to reach the blue market point ahead, the robot must avoid a collision. The path chosen by the agent is shown by the dotted line.

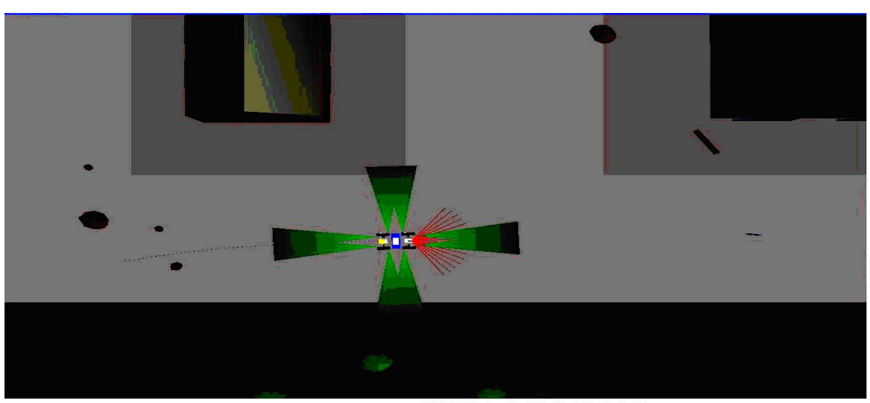

Fig. 2 - Reaching the goal, after applying obstacle avoidance behavior

5. Conclusion

In this paper a fuzzy based methodology for robot navigation in previously unknown environments is presented. Results show that the robot achieves the goals received and follows a free of collision path. As all experiments were done in a simulated environment future work aims for usage of a real robot in real-time situations.

Received April 15th 2010

* *"Gh Asachi" Technical University of Iassy*
Theory of Mechanisms and Robotics Dep.
e-mail: rozmarincristina@yahoo.com
** *Royal Military Academy of Brussels*
Department of Mechanics
e-mail: sidahmed.berrabah@rma.ac.be

R E F E R E N C E S

1. A l l e s a n d r o S a f f i o t t i, *Autonomous Robot Navigation: a fuzzy logic approach*, IRIDIA, 1998.
2. D o u g l a s S. B l a n k and J. O l i v e r R o s s, *Learning in a Fuzzy Logic Robot Controller*, Proceedings of the 1997 Meeting of the American Association of Artificial Intelligence
3. J a n J a n z e n, *Tutorial on fuzzy logic*, Technical University of Denmark , 1998
4. P a n a g i o t i s G. Z a v l a n g a s, P r o f. S p y r o s G. T z a f e s t a s, D r. K. A l t h o e f e r, *Fuzzy Obstacle Avoidance and Navigation for Omnidirectional Mobile Robots*, ESIT 2000, Aachen, Germany, 14-15 September 2000
5. P i e r r e Y v e s G l o r e n n e c, *Reinforcement Learning: An Overview*, INSA de Rennes / IRISA
6. R e i n h a r d B r a u n s t i n g l, J o k i n M u j i k a, J u a n P e d r o, J. M. A r i z m e n d i a r r i e t a U r i b e, *A Wall Following Robot with a Fuzzy Logic Controller Optimized by a Genetic Algorithm,* FUZZ-IEEE /IFES '95 FUZZY ROBOT COMPETITION, YOKOHAMA, P.77-82, MARCH, 1995
7. S. T h o n g c h a i, S. S u k s a k u l c h a i, M. W i l k e s, N. S a r k a r, *Sonar Behavior – Based Fuzzy Control for a Mobile Robot,* School of Engineering, Vanderbilt University, Nashville, TN 37235

CONTROLUL UNUI ROBOT MOBIL ÎNTR-UN MEDIU SIMULAT FOLOSIND LOGICA FUZZY ŞI ÎNVĂȚAREA CU ÎNTĂRIRE

Lucrarea de față prezintă o metodologie bazată pe Logica Fuzzy şi Învățarea cu Întărire pentru controlul unui robot mobil într-un mediu simulat. Obiectivul urmărit este deplasarea spre un anumit punct şi obținerea unei traiectorii fără coliziuni.