Robot Navigation Based on Adaptive Fuzzy Controller

Sid Ahmed Berrabah, Cristina Gabriela Rozmarin

Mechanical Department, Royal Military School, Belgium,
Avenue de la Renaissance 30, 1000 Brussels, Belgium
sidahmed.berrabah@rma.ac.be

I. Abstract:

This work introduces a sensor-based methodology for robot navigation in unknown environments where a fuzzy logic system translates the sensor measurements directly to actuator actions. The approach uses a behaviorist design and a strategy for multi-behavior coordination. Simple fuzzy control is suitable to design a simple behavior as to reach a goal. However, it is difficult to maintain the correctness, consistency, and completeness of the fuzzy rule base for an obstacle avoidance behavior constructed and tuned by a human expert. Therefore, a fuzzy system able to evolve and automatically improve its performance is used in our application. The learning is realized by an unsupervised approach called reinforcement learning.

The navigation process is adapted for the ROBUDEM robot, implemented as CoRoBA module and tested using the CoRoSim simulation platform.

II. Introduction

Robot navigation is a field of research which is been evolving for a few decades now (see for example [1]), yet the amount of concepts only seems to increase with time, instead of leading to a clear definition of a superior method. The reason for this presumably is that almost every task / environment / robot has its own strategy which is most fit for the job. Yet, in general, there is one clear division one can make between al the solutions presented. On the one hand we have methods where some kind of a map is built and the path planning procedure uses this map to find a safe passage to a goal point. On the other hand there are the methods which do not rely on any kind of map and where the path planner is able to generate a solution directly based upon the sensor readings. The latter technique relies nowadays most often on soft computing techniques like fuzzy logic and neural networks. Examples of this approach can be found amongst many others in [2] and [3].

In this paper, we present a sensor-based methodology for robot navigation in unknown environments where a fuzzy logic system translates the sensor measurements directly to actuator actions. The approach adopts a behaviorist design and a strategy for multi-behavior coordination. Simple fuzzy control is suitable to design a simple behavior as to reach a goal. However, it is difficult to maintain the correctness, consistency, and completeness of the fuzzy rule base for an obstacle avoidance behavior constructed and tuned by a human expert. Therefore, a fuzzy system able to evolve and automatically improve its performance is recommended. In this application, the unsupervised learning approach (reinforcement learning) is used.

The navigation process is adapted for the ROBUDEM robot, implemented as CoRoBA module and tested using the CoRoSim simulation platform.

III. Fuzzy Logic

The concept and terminology of fuzzy logic were first introduced by Lotfi Zadeh. Fuzzy logic is an extension of the crisp logic to provide a mathematical interpretation for natural language terms where little ambiguity exists. In our daily lives, there exist countless vague concepts that we humans can easily describe, understand and communicate with each other; but conventional mathematics, including the set theory, fails to handle this in a rational way. The extension lies in the fact that a fuzzy set may contain its elements partially, whereas an element of a classical set either belongs to the set or does not. In a fuzzy set S, each element x of the set is assigned with a degree of membership in S, which is measured by a membership function $\mu_S(x){:}R \mapsto [0,1]$. The membership function $\mu_S(x)$ is zero when x does not belong to S at all, one when x belongs to S totally and $0 < \mu_S(x) < 1$ when x belongs to S partially.

In the literature, the most popular application in the fuzzy logic domain are fuzzy controllers.this type of controller is used to control complex processes when no precise model of the process exists and most of the information is available only in qualitative form.

III.1. Fuzzy Controllers

The purpose of the fuzzy logic controller is to compute values of action variables from observation of state variables of the process under control.

Figure 1 shows the basic configuration of a fuzzy logic controller, which comprises four principal components [4]: the fuzzification interface, the knowledge base, the decision-making logic, and the defuzzification interface.

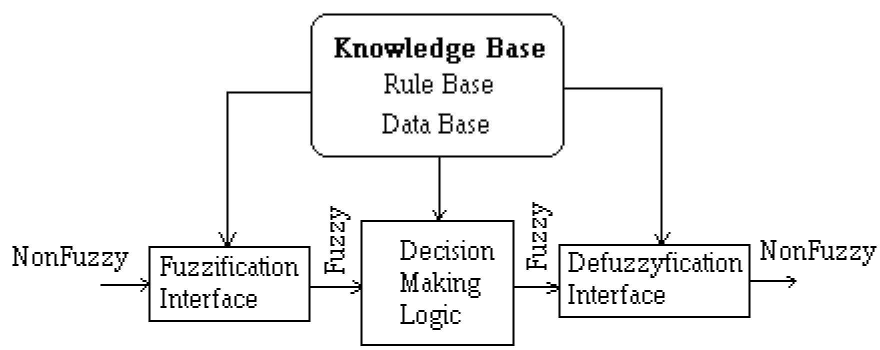

Figure 1: Configuration of a Fuzzy Logic Controller

The first component is the fuzzification interface, which performs the function of fuzzification that converts input data into linguistic values, which may be viewed as labels of fuzzy sets.

The second component is the knowledge base, which comprises the database and the rule base. The essential part of the fuzzy logic controller is a set of linguistic rules such as:

$$R^l\text{: if } x_1 \; is \; {X_1}^l \wedge \ldots \wedge x_n \; is \; {X_n}^l \text{ then } y \; is \; Y^l \quad (1)$$

The rule's antecedent part contains the controlled variables, while the consequent part contains the controlling variables. ${X_1}^l$ and Y^l are fuzzy sets in the input and the output universe of discourse respectively. Practice has shown that these fuzzy IF-THEN rules provide a convenient framework to incorporate human experts' knowledge. Each fuzzy IF-THEN rule defines fuzzy sets ${X_1}^l \times \ldots \times {X_n}^l \rightarrow Y^l$ in the product X × Y.

The database provides necessary definitions used to characterize the fuzzy control rules and the fuzzy data manipulation in the fuzzy logic controller.

The third component is the decision-making logic which simulates human decision-making. It consists in calculating the membership function for the consequent from all activated rules for the given inputs. The most commonly used fuzzy inference engine is the so-called Max-Min composition in which, the output determined by all fuzzy IF-THEN rules is a fuzzy set whose membership function is

$$\mu_y = \max_i \left(\min \left(\mu_{X_1^i}(x_1) \cdots \mu_{X_n^i}(x_n) \ , \ \mu_{y^i}(y) \right) \right) \tag{2}$$

The last component of the decision process is the defuzzification interface, which computes a non-fuzzy control action from an inferred fuzzy control action. There are different methods in order to realize the defuzzification. There is not any procedural method to choose which method is more suitable. Most common used methods are : middle of maximum, centroid, largest of maximum. In the centroid method the output result is calculated by:

$$y = \frac{\int \mu_i \, y^i}{\int \mu_i} \tag{3}$$

Where μ_i is the overall truth value of the premise of rule R^i for the input.

III.2. Takagi and Sugeno's Fuzzy Controller

In the controller proposed by Takagi and Sugeno the consequent part of the fuzzy IF-THEN rule is linear combination of the input variables:

$$R^l\text{: if } \ x_1 \ is \ {X_1}^l \ \wedge \ldots \wedge \ x_n \ is \ {X_n}^l \quad \text{then} \quad y^l = {c_0}^l + {c_1}^l x_1 + \ldots + {c_n}^l x_n \tag{4}$$

Where ${c_i}^l$ are real-valued parameters. The output in this case is a weighted average of the y^l's

$$y = \frac{\sum_i \mu_i \, y^i}{\sum_i \mu_i} \tag{5}$$

As said above, fuzzy controllers are usually designed by expressing in the rule base the way that experts make decisions. However, this natural extraction of experts'a priori knowledge is not always easy or possible to realize, especially for the conclusions of the rules [5]. Indeed, problems can come from disagreements between experts'decision rules which are difficult to structure, or due to a great number of variables necessary to solve the control task. For all these reasons, many research are being made to find automatic methods allowing self tuning of fuzzy inference systems. Among these methods, we can cite the use of neural networks [6], genetic algorithms [7], adaptive algorithms[5], reinforcement learning [8][9].

III.3. Adaptive Fuzzy Controller using reinforcement learning

Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions an agent ought to take in those states [10][11]. The environment, in return, provides a reward which can be positive or negative. The reinforcement learning algorithm consists to find a policy for maximizing cumulative reward for the agent over the course of the problem.

In our application the agent is the mobile robot. At any time step, the robot should take a decision to avoid the obstacles or to reach it goal. When the decision is taken and the action performed, the decision process receive a feedback in the form of a reward from the environment which indicates if this is a good or bad decision to take in attempting to achieve the goal. As the learning concerns only the obstacle avoidance controller, the action correspond to the possible scalar outputs of the controller and the states correspond to the different combinations of the input fuzzy variables of the controller.

The used algorithm for reinforcement learning of the fuzzy controller is the Q-algorithm, which can be summarized as follows:

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.

1. Return to 2

IV. Path Planning Algorithm

The proposed path planning procedure is based on two fuzzy logic controllers: a goal seeking controller and an obstacle avoidance controller. The goal seeking controller tries to find the optimal path to the intermediate goals (defined by the global path planning), while the obstacle avoidance controller has for mission to avoid obstacles. A command fusion scheme based on fuzzy operators arbitrates between the two behaviors.

IV.1. Goal Seeking Controller

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

IV.2. Obstacle Avoidance Controller

Given the data received from the sonar sensors and the direction to goal, the Obstacle Avoidance Controller (OAC) calculates the turning angle and the velocity of the robot in order to avoid obstacles. The fuzzy logic controller is also a "Sugeno" type.

The sensors are grouped in five groups: Front Right, Front Left, Lateral Left, Rear, and Lateral Right.

The tuning of the OAC is done using the online Q-learning algorithm. The parameters of the OAC are adjusted based on the reward parameter as described above.

V. Experiments and Results

The Validation of the proposed navigation algorithm was using a Java based multi mobile robot simulator MoRoS3D. A simulator as such increases safety when developing and testing algorithms. In MoRoS3D a robot can be placed in a 3D environment and interact with that environment in a manner similar to that of the robot in the real physical situation. Although MoRoS3D visualizes the entire surroundings of the robot, the robot software only ”sees” the information it collects through its sensors, just like with a physical robot. The MoRoS3D simulator provides simple interaction with the user and offers different virtual cameras including on-board and tracking ones. Simple distance sensors, such as Laser, US and IR, are simulated.

Figure 2 shows an example of the developed path planning system in some realistic cases. The path of navigation is shown with dotted line.

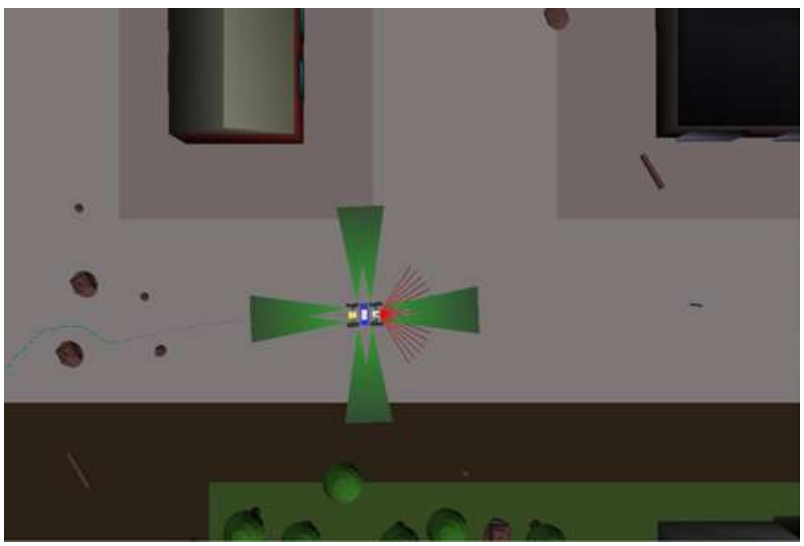

Figure 2: Examples of navigation in realistic environments.

VI. Conclusion

We presented an algorithm for mobile robot navigation in previously unknown environments. The proposed algorithm is based on adaptive fuzzy logic controller.
The use of the simulator MoRoS3D has helped the development and tuning of the algorithm but as perspective of this work, is to use of the proposed navigation algorithm in real applications using real robot.

VII. References

[1] N.J. Nilsson: A Mobile Automaton: An Application of Artificial Intelligence Techniques, Proceedings of the 1st International Joint Conference on Artificial Intelligence, Washington D.C., pp. 509-520, 1969.
[2] P.S. Lee, L.L. Wang, Collision Avoidance by fuzzy logic control for automated guided vehicle navigation, Journal of robotic systems, Vol. 11, Nr. 8, pp. 743-760, 1994
[3] R. Glasius, A. Komoda, S. Gielen, Neural network dynamics for path planning, Neural Networks, Vol. 8, Nr. 1, pp.125-333, 1995.
[4] L.X. Wang, Adaptive Fuzzy Systems and Control,PTR Prentice Hall,1994.
[5] B.Kosko, Neural Networks And Fuzzy Systems, Prentice Hall,1992.
[6] W.S. Lin and C. H. Tsai, Robust neuro-fuzzy control of multivariable systems by tuning consequent membership functions, Fuzzy Sets and Systems, vol. 124, pp.181-195, 2001
[7] Castillo L., A. González, R. Pérez, Including a Simplicity Criterion in the Selection of the Best Rule in a Genetic Fuzzy Learning Algorithm, Fuzzy Sets and Systems, v.120 (2) , pp.309-321,2001.
[8] Cang Ye, Nelson H. C. Yung, Danwei Wang, A Fuzzy Controller With Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance, IEEE Trans on Systems, Man and Cybernetics, Part B, pp1083-4419,2002.
[9] Dong-Oh Kang,Zeungnam Bien, Design of Multi-objective Satisfactory Fuzzy Logic Controller using Reinforcement Learning, International Journal of Fuzzy Systems, Vol. 2, No. 2, pp139-152, June 2000
[10] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research, 1996, Vol.4, pp.237-285.
[11] R. S. Sutton, A.G. Barto, Reinforcement Learning: in introduction, (online book) MIT Press, 1998.