The Comparative Application of Fuzzy Logic and Genetic Algorithm for Intelligent Navigation of Mobile Robot in Dynamic Unknown Environment in the Case of Fixed and Moving Obstacles

Ali Reza Naderloo^{*1}, Fazlollah Adibnia¹ , Alimihammad Latif¹

1- Department of Software Engineering, Islamic Azad University Yazd Branch, Yazd- Iran

ABSTRACT

Introduction: In recent years, topics related to robotics have become one of the areas of research and development. In the meantime, smart robots are very popular, but the control and navigation of these devices are very difficult, and the lack of handling and staggering obstacles and avoidance of them, due to safe and secure routing, outweigh the basic needs of these systems.

Material and Methods: In this research, for the purpose of solving the intelligent navigation problem, a moving robot in a dynamic unknown environment (conditions at any moment in the range of moving and obstacles in motion) and the choice of optimal path, the methods of genetic algorithm and fuzzy logic are used comparatively.

Results: By using genetic algorithm and fuzzy logic methods, the robot can move in the dynamic and unknown environment to the optimal path to the target.

Conclusion: Information about the environment is also necessary to avoid obstacles, optimal path design and environment exploration, and to establish a clever relationship between perception and practice that requires the use of appropriate algorithms such as the genetic algorithm and fuzzy logic (fuzzy controller) are also needed to manage the control and navigation.

Article History

Received: 2016-03-12

Accepted: 2016-06-31

Published: 2016-07-13

Keywords

Smart Control

Navigation

Avoidance of Barriers

Optimal Routing

Genetic Algorithm

Fuzzy Logic

①Please cite this article as:

Naderloo AR, Adibnia F, Latif A. The Comparative Application of Fuzzy Logic and Genetic Algorithm for Intelligent Navigation of Mobile Robot in Dynamic Unknown Environment in the Case of Fixed and Moving Obstacles. Iran J Med Inform. 2016; 5(1): 28-34. DOI: [10.24200/ijmi.v5i0.102](https://doi.org/10.24200/ijmi.v5i0.102)

* Corresponding Author: AR Naderloo, Department of Software Engineering, Islamic Azad University, Yazd Branch, Yazd, Iran (Email: alir_naderloo@yahoo.com)

INTRODUCTION

Robots must be able to pursue their targets in the face of unexpected changes in the environment for operations in unrecognized real-life environments. Real environments are rarely predictable or well known; therefore, the creation of a precise, accurate movement path, before moving in such environments, will not generally be applied [1]. In general, the problem of designing a robot's path can be broken down into two sub-issues [2].

1. Following Goal: Because short paths to the target cannot be achieved only with local information; therefore, the topology (space) of space is important in finding the right paths to the goal.
2. Avoiding Obstacles: It often only solves using local information, which means that the robot must first feel the barriers to avoid them. The robot moves in environments that are often unknown, so the robot cannot be scheduled to do the foreground. The robot should be equipped with precision sensors to identify the environment and prevent collisions with objects.

In general, we can categorize the solutions provided for routing design and robot motion problems in the form of two subsets of "classical methods" and "heuristic methods" [2].

Classical methods

These methods, in general, come from a combination of several general methods, such as road mapping, Cell Decomposition, Artificial Potential Field (APF), and Mathematical Programming. Most of the routing and robotic design issues can be solved with these methods. In the roadmap method, a free configuration space (the configuration space is a collection of all the configurations that the robot can achieve, and the free space configuration is a set of configurations that do not interfere with the barriers. Then this space with the number of degrees of freedom of the robot being determined) in which the movements are shown with a network of one-dimensional lines. The search for a solution to this network is limited to turning the routing and robot motion planning into a geographic search problem. The most commonly used graph is the Graph Clearly.

In this way, a network of lines connects a feature of objects that are usually polygon vertices. An unstoppable route is found through this network. In the method of cellular algorithm, the free configuration space is decomposed into a set of simple cells and the relationship between adjacent cells is calculated. Then a pathless collision with the connection of the start and the target cells with the sequence of the related cells is created between the start and target configurations. In the method of artificial potential fields, the basis is the use of an artificial potential field, which consists of a strong repulsion force adjacent to the obstacles and a gravity force generated at the target position. The integration of these two forces creates a potential field that gives information about the environment. Using this information, one can find a path that directs the robot to the target position by avoiding obstacles. In mathematical programming, the problem of routing projection and robot movement is transformed into a mathematical optimization problem, which ultimately results in a curve between the start and target configurations by minimizing a certain scalar quantity.

Heuristic methods

Heuristic methods are included Path Velocity Decomposition (PVD), Accessibility Approach, Probabilistic Approach and Relative Velocity Diagram. In classical ways, there are problems like computational complexity in higher dimensions of configuration space, as well as high complexity problems. In addition, some solutions, such as the method of artificial potential

fields, are local, which does not allow the robbery to occur in the local minimum points (local minimum points are those points that are slightly larger than the minimum points and are near the path) [3]. Hence, modern heuristic methods are proposed to improve their performance.

MATERIAL AND METHODS

Many of the routing methods described in the previous section are often not sufficiently flexible in conditions that block a robot path to the target (using predefined functions) and have problems with time complexity. In this research, we have tried to apply modern methods such as fuzzy logic (because for robot routing in unknown environments in the presence of fixed and moving obstacles, without knowing the position and function of moving obstacles and momentarily and knot to the node Look for the momentary routing methods) as well as the genetic algorithm method (since the genetic algorithm with a set of values of decision variables, each of which (equivalent to a chromosome) can be the answer to the plan, in parallel), while other methods are limited to just a set of decision variables, which are just one possible answer for the designer They continue to search for work); and given the uncertainty of the input information, the routing planning and robotic motion plan was designed optimally and intelligently. The following are also noteworthy about the configuration space of a moving robot [4, 5].

- For a mobile robot, we are planning to consider it as Holonomic. In this case, the robot can be considered as a point.
- As a result, the position space can be represented as two-dimensional with axes x, y.
- In this case, the objects in the environment are as large as the robot's radius, so that the robot's point of view is correct.

In Fig 1, the position space is shown as a two-dimensional moving robot.

Fig 1: Two-dimensional moving robot

Using fuzzy logic to navigate the smart robot in animated dynamic environments

The fuzzy term in Oxford's dictionary is defined vague, dumb, inaccurate, confusing, confusing, indeterminate, and indeterminate. In this text there are two justifications for the theory of fuzzy systems [6]:

First justification: Our real world is much more complicated than being able to obtain an accurate description and definition, so an approximate description, that is, an acceptable, analysis, should be introduced for a model.

Second justification: We need a hypothesis that can systematically formulate human knowledge and put it along with other mathematical models in engineering systems.

In fuzzy logic, we work with non-deterministic and approximate values; a range of probabilities that may occur. When a fuzzy system is used as a controller, it is referred to as a fuzzy controller [7]. In Fig 2, the structure of a fuzzy controller is displayed.

Fig 2: The structure of a fuzzy controller

In this section, we need to go to the routing methods for robot routing in unknown environments in the presence of fixed and moving obstacles, without knowing the position and movement function of the obstacles and as nodes to the node. Fuzzy logic has very good performance in point-to-point routing and node-to-node [7]. Routing from the first point is done as a node to the node until the robot reaches the target point. With this method, at each moment, the next node, with 8 options in this study, is considered to be 8-way. Available using fuzzy logic is used according to the Fig 3.

Fig 3: Available moves to move towards new nodes

The fuzzy inputs that are considered here for routing are:

- The difference between the angles of the next nodes to the target
- Distance of next nodes to the nearest obstacle

It should be noted that at any moment, at each node in which the robot is positioned, it will only know the position of the obstacles, just up to a certain radius determined by the robot sensor board [8, 9]. For example, if the sensor board is 30 cm (within the simulation program, it is assumed that this radius is 30 cm), at any moment the robot is aware of the barriers to a radius of 30 cm around it.

At any given time, routing and selecting the next node are made from among the eight options available (of course, it should be explained that you can select up to 16 modes or even more). For each of these 8 nodes, a preference factor is selected as the next node (using fuzzy logic). After executing fuzzy logic (performing fuzzy operation) for each node's inputs, an output that indicates the priority of that node's selection is created. Finally, the node is the node with the highest priority factor. The general stages of node-to-node routing are done using fuzzy logic as Fig 4.

Fig 4: General stages of optimal routing to output generation, by fuzzy logic [9]

Fuzzification of the input

Navigation via the fuzzy logic is done node to node and momentarily. In fact, the fuzzy logic is done as frequently as the number of nodes along the way [10]. Two fuzzy inputs have been used for the navigation purpose in this system: angle difference with the target destination and the distance from the nearest obstacle. For each input, 5 membership functions have been used for fuzzification. The membership functions have been taken here as hypothetical). In Fig 5 and 6, we can see the membership functions for the angle input and the distance input.

Fig 5: the membership functions for the angle input [10]

Fig 6: the membership functions for the distance input [10]

It is noteworthy that momentarily the non-fuzzy input for each next 8 nodes get normalized and reported as 0-1. Only then, the fuzzification follows. In this system, for the output, 5 membership functions are used as the following [10]:

- Very Low (VL)
- Low (L)
- Moderate (M)
- High (H)
- very High (VH)

The overall procedure begins with the estimation of the shortest distance of every node to the nearest obstacle as well as the angle difference to the target destination normalized between 0 and 1. As the next step, for each and every node, the normalized non-fuzzy input gets fuzzified from the nearest existing obstacles as well as the angle difference from the target destination using the relevant fuzzy membership functions. The membership functions for the fuzzy output is as in Fig 7.

Fig 7: The membership functions for the fuzzy output [10]

Table of the fuzzy rules:

As mentioned earlier, each next node is selected in the pathway through the fuzzy logic. The fuzzy rules for selecting the next node in momentary navigation are indicated in Table 1. According to this table, the more the distance from the nearest obstacle and the lower the angle difference from that of the target, the higher the probability of that node to be selected as the next node along the pathway.

Table 1: The fuzzy rules for selecting the next node in momentary navigation

Output: the higher the probability of that node	The lower the angle difference from that of the target	The distance from the nearest obstacle
(M)	(VL)	(VL)
(L)	(L)	(VL)
(VL)	(M)	(VL)
(VL)	(H)	(VL)
(VL)	(VH)	(VL)
(M)	(VL)	(L)
(M)	(L)	(L)
(L)	(M)	(L)
(L)	(H)	(L)
(VL)	(VH)	(L)
(H)	(VL)	(M)
(M)	(L)	(M)
(M)	(M)	(M)
(L)	(H)	(M)
(VL)	(VH)	(M)
(VH)	(VL)	(H)
(VH)	(L)	(H)
(H)	(M)	(H)
(M)	(H)	(H)
(L)	(VH)	(H)
(VH)	(VL)	(VH)
(VH)	(L)	(VH)
(H)	(M)	(VH)
(M)	(H)	(VH)

Output: the higher the probability of that node	The lower the angle difference from that of the target	The distance from the nearest obstacle
(M)	(VH)	(VH)

To find the fuzzy output in each node, reference can be made to the table of fuzzy rules which contains all the rules as AND (hypothetical rules). In the next step, the fuzzy output can be conceptualized as a fuzzy output through Mamdani's multiple method or Max-Min composition method [2]. In this step, one or more of the rules within Table 1 are implemented in accordance with the fuzzy input of each node.

Defuzzification and setting the final priority coefficient

Defuzzification methods are used in this paper is shown in fig 8.

Fig 8: Defuzzification methods

As mentioned earlier, the most common defuzzification methods are the Center of Gravity (COG) and the Center of Average (CA) method [9]. To design the present nanorobot navigation system, the latter method was used for defuzzification. Eventually, once the defuzzified output is determined from among the existing 8 options, the option with maximal defuzzified output is selected as the next node along the pathway. The above procedure goes on until the robot reaches the target destination. Fig 9 includes how the optimal path is decided upon and how the mobile and immobile obstacles are faced with in 6 hypothetical steps.

Fig 9: Optimal path decision

Application of genetic algorithms for intelligent mobile robot navigation in unknown dynamic environments

The scope of the genetic algorithm is vast and every day, with the ever-increasing advances in science and technology, this method has greatly expanded to optimize and solve problems. Genetic algorithm is one of the subset of evolutionary calculus, which has a direct relation with the topic of artificial intelligence. In fact, the genetic algorithm is one of the sub-sets of artificial intelligence [11].

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm

repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

For robot routing in unknown environments, in the presence of fixed and moving obstacles, without knowing the position and function of the obstacle movements, the genetic algorithm can also be helped, and given the variable path length, the structure of the variable chromosome (Due to the fact that different paths have varying lengths, the structure of variable-length chromosomes has been used) [12].

For robot motion routing, variable length chromosome structure is used due to the length of the path. Thus, each gene of number 1 expresses the midpoint along the path (the path determined by each chromosome). In other words, the first gene always represents the starting point and the last gene represents the target point. Initially, a randomized population of chromosomes is created. After evaluating the solutions produced in each replication, some of the best chromosomes are selected as parents for the next generation. Then, the process of combining and mutation on the parent chromosomes generates the next-generation chromosomes. The process continues to the extent that the termination condition is satisfied. The steps of routing to output will be executed as shown in Fig 10.

Fig 10: The stages of routing operations to output by Genetic algorithm

To generate initial solutions, each chromosome acts as a nodal node for generating a possible pathway, so that at each step, a node from among the eight options in this study is considered to be a pathway of 8. Existing times are based on the law of probability of choice and complete their paths and ultimately leads to a solution. This selection is done according to equation (1). Then the population of the initial answers is selected.

$$P(t_i) = \begin{cases} \frac{\left(\frac{1}{h_i}\right)^\delta}{\sum_{j=1}^N \left(\frac{1}{h_j}\right)^\delta} & \text{if } t_i \notin \text{Blacklist} \\ 0 & \text{if } t_i \in \text{Blacklist} \end{cases} \quad (1)$$

In this research, we considered the population of the genetic algorithm to be equal to 50 and the number of iterations equal to 100 (of course, hypothetically). To achieve the desired accuracy, we evaluated the genetic algorithm several times with different parameters and operators. Finally, the best values of the parameters. We chose the best operators for the algorithm: the probability of mutation for each gene was 0.05% and the probability of the combination was 90%. As discussed earlier, here the selection method in the genetic algorithm is also the method of choosing elitism. Initially, the robot's initial position in the coordinates [5, 50] and the target position in coordinates [450, 470], the result of the final simulation using the genetic algorithm is shown in Fig 11 and 12 compares the optimal routing between fuzzy logic and genetic algorithm methods using the same initial points and endpoints.

Fig 11: Routing using a genetic algorithm in a complex environment in several successive steps (path length 742.9037 cm)

Fig 12: Comparison of routing between fuzzy logic (right) and genetic algorithm (left side), with the same points at the beginning and the end

RESULTS

By conducting this research and comparing fuzzy logic and genetic algorithm methods, it was determined:

1. The research system under the conditions of using the fuzzy logic method for robotic routing [13]:

- No need for prior knowledge of the environment (very useful for work in unknown environments).

- At any moment, only the perimeter of its environment (in particular, in the design, up to 30 cm from each side) could be identified (very useful in terms of time cost and somewhat useful for optimization), in fact at any moment the position The current obstacle (fixed and moving) was up to 30 centimeters. The system was also informed of the target's coordinates.
- It was easy to understand and solve the problem.
- The combination of input parameters (the difference in the angle of the next nodes to the target and the distance between the next nodes and the nearest barrier) was carried out simply and easily.
- Control and adjustment of output and output parameters were easily accomplished with the help of fuzzy rules.

2. Research system in terms of using the genetic algorithm for robotic routing [13]:

- Firstly, in an evolutionary process, they design the route (recognizing the path) and then issue a robotic command (high resource use)
- Every time before moving, they should check the position of fixed and moving obstacles (somewhat inefficient in terms of cost and time-consuming in terms of optimization) and all of the search space in parallel.
- Highly efficient for solving discrete and nonlinear problems (probabilistic methods), such as routing robotics.
- Strong and flexible as a search method.
- A very good analysis of the combined (location-time) issues and has local opticians.

They used a large number of scattered points to conduct searches.

DISCUSSION

In the same conditions, in most experiments, the time spent (in seconds) is "shorter" than the time spent using the genetic algorithm to travel the optimal path in unknown environments using fuzzy logic. Under the same conditions, the distance traveled (in cm), in order to follow the optimal path in unknown environments using the genetic algorithm, the "shorter" distance from the fuzzy logic has been studied in this study. : In terms of time scale, the fuzzy logic method, in the optimal routing for unknown environments, takes a shorter time; also the spatial scale and distance traveled, using a genetic algorithm, the optimal routing for unknown environments, can bring better conditions [14].

Considering the results obtained from the design implemented in this study and the comparisons made, in order to better address these issues, the suggestion to use the "fuzzy hybrid hybrid system" to optimize the parameters of the fuzzy system using the genetic algorithm " In fact, in this method, it is suggested that the routing be done instantaneously using fuzzy logic (ie, optimizing the sample path in terms of the time it takes to work, in its shortest time) (at this time, only One time, non-momentarily and online, the genetic algorithm used to adjust the various parameters used In the fuzzy system [8, 5].

CONCLUSION

The present findings from the implemented designs here took into account the merits and demerits and evaluated the optimal

navigation issue in robots following the fuzzy logic and the genetic algorithm. We found out that to solve such issues more efficiently, we suggest using an integrated system of genetic and fuzzy algorithms (optimizing the parameters of a fuzzy system via the genetic algorithm). In fact it is suggested in this approach that the navigation be done momentarily using the fuzzy logic. In other words, optimizing the sample pathway takes the least time possible. Meanwhile, we can only once and non-momentarily use the genetic algorithm offline to set the different parameters in a fuzzy system. As an instance, genetic algorithm can be used for the optimal setting of the membership functions used in the fuzzy logic as well as the optimal setting of the fuzzy logic table of rules. This would provide for all the advantages of the fuzzy logic simultaneously in optimizing the Nano robot therapists' navigation.

In other words, if a combination of the genetic and fuzzy systems is used we can optimally benefit from the two main factors involved in moving along the pathway in the least time possible and through the shortest distance. Finally, an ideal and optimal path will be found in terms of the main factors of time and distance for the target navigation.

REFERENCES

1. Ratering S, Gini M. Robot navigation in a known environment with unknown obstacles. *Auton Robots*. 1995; 1(2): 149-65.
2. Boada MJL, Egido V, Barber R, Salichs MA. Continuous reinforcement learning algorithm for skills learning in an autonomous mobile robot. *Proceeding of IEEE International Conference on Industrial Electronics Society*. Sevilla, Spain. 2002.
3. Borenstein J, Everett B, Feng L. *Navigating mobile robots: Systems and techniques*. AK Peters Ltd., Wellesley, MA, 1996.
4. Lamiroux F, Bonnafoous B, Lefebvre O. Reactive path deformation for non-holonomic mobile robots. *Robotics*. 2004; 20(6): 967-77.
5. Qu Z, Wang J, Plaisted CE. A new analytical solution to mobile trajectory generation in the presence of moving obstacles. *Robotics*. 2005; 20(6): 978-92.
6. Vichuzhanin V. Realization of a fuzzy controller with fuzzy dynamic correction. *Central European Journal of Engineering*. 2012; 2 (3): 392-8.
7. Hagrass HA. A hierarchical type two fuzzy logic control architecture for autonomous mobile robots. *Fuzzy Systems*. 2004; 12(4): 524-39.
8. Beom HR, Cho HS. A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *Systems, Man, and Cybernetics*. 1995; 25(2): 464-77.
9. Gerla G. Fuzzy logic programming and fuzzy control. *Studia Logica*. 2005; 79(2): 231-54.
10. Ye C, Yung NHC, Wang D. A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance. *Systems, Man, and Cybernetics*. 2003; 33(1): 17-27.
11. Mitchell M. *An introduction to genetic algorithms*. MIT Press, Cambridge, MA, 1996.
12. Goldberg DE. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
13. Petrenko YN, Alavi SE. Fuzzy logic and genetic algorithm technique for non-linear system of overhead crane. *Proceeding of International Conference on Computational Technologies in Electrical and Electronics Engineering*. 2010.

14. Vafaie H, Imam I. Feature selection methods: Genetic algorithms vs. greedy-like search. Proceeding of the International Conference on Fuzzy and Intelligent Control Systems. 1994.