Development and Analysis of a Fuzzy Controller for Mobile Robots in Heterogeneous Soils

Israa M. Abdalameer Al-Khafaji

***Abstract*— This paper explores using fuzzy logic to control diverse-soil mobile robots, aiming to create an adaptable system for swift environmental changes. Fuzzy51 software showcases system design and performance on various soils. Discoveries include system modeling, analysis within Fuzzy51, knowledge base, fuzzy control logic design, and strategy discussions. The fuzzy controller, with or without memory, adeptly handles environmental changes, ensuring accurate decisions. It details the use of the fuzzy control tactics manipulator, including strategy settings and a system for blocking specific exchanges. The study examines inverter and repeater functions before and after X = 50, presenting simulation outcomes. In summary, it proposes regulating robots in varied soils via fuzzy logic, showcasing adaptability to rapid environmental changes.**

***Keywords*— Fuzzy Knowledge Bases (FKBs), fuzzy controller (FC), Fuzzy logic, Robotics, Comparison, Linguistic Input Variable X1, System Analysis.**

I. INTRODUCTION

Robots are electromechanical machines that execute tasks independently depending on their programming and instructions. Manufacturers and makers of robots aspire to construct machines that are as accurate as people and capable of doing difficult, repetitive, dangerous, and tedious activities that humans want to avoid. Robotics technological advancements, such as the usage of microcontrollers and microprocessors, have allowed robots to detect their surroundings, make intelligent judgements, and react to emergency circumstances, marking a key milestone in the robotics industry.

A. Application of Artificial Intelligence Technologies in Robot Control:

To improve the control and efficiency of mobile robots, AI technologies such as neural networks, reinforcement learning, genetic algorithms, and fuzzy logic may be used. Fuzzy logic, in particular, excels in dealing with emergency situations and making suitable conclusions. Fuzzy control systems depend on knowledge bases made up of fuzzy rules generated by domain experts. Fuzzy Knowledge Bases (FKBs) enable fuzzy systems to accomplish tasks without explicit computations and measurements, making them popular among academics and scientists [1].

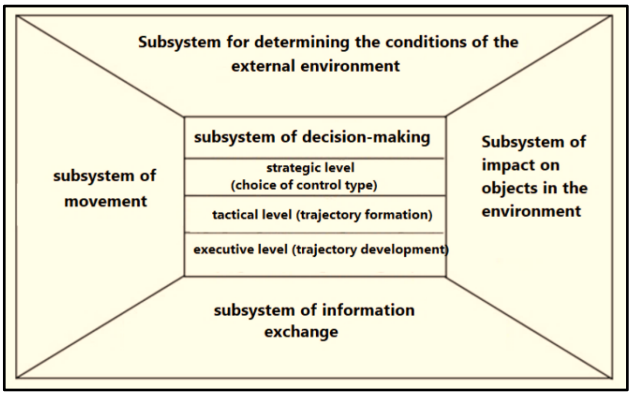


Figure 1.1 illustrates the generalized structure of a modern robotic system [2].

B. Progress in Practical Robotics:

Practical robotics applications have lately made great development. A Fuzzy Knowledge Base (FKB) is essential in fuzzy control for maintaining control objects at the tactical level of control. A system often includes multiple object management activities. While one FKB may solve a single issue well (other problems may have differing degrees of effectiveness) or exist for a certain set of conditions, reaching optimum problem-solving needs numerous FKBs for different situation sets. However, a single FKB is not generally applicable, and it is preferable to use a number of unique and smaller FKBs. The selection of one or more FKBs from this collection presupposes the existence of a dynamic FKB. The goal of this effort is to provide a system that allows for the dynamic selection of FKBs.

II. MAIN RESULTS OF THE STUDY

System analysis, a set of fuzzy knowledge bases, a control object, and a fuzzy controller are the four main building pieces of a system architecture for developing fuzzy logic modelling systems, as shown in Figure 2.1.

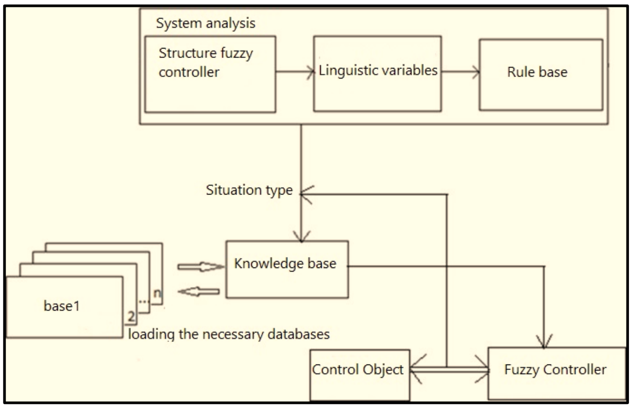


Fig. 2.1. Structure of modeling systems.

A. System Structure and Principles of Operation:

The system structure consists of four components, and we will go through how they work. The first component is "System Analysis," which entails thinking about things like circumstances, issue type, and variables used. System analysis is the first stage in understanding the complexities of a system, especially a simulation system in this instance. It includes establishing input and output values, circumstances, effects, and comparing to other systems. Figures 2.2 and 2.3 provide detailed explanations of the construction and analysis of the modelling system, using the Fuzzy51 system as an example. This examination covers the structure of the Fuzzy Controller (FC), linguistic variables, and the rule basis [3].

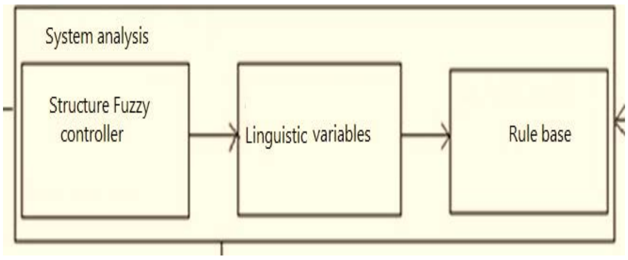


Fig. 2.2. Block "System Analysis".

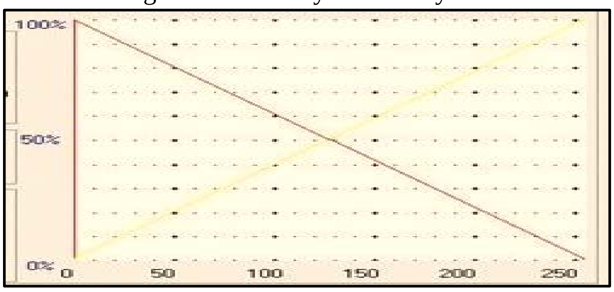


Fig. 2.3. An example of system analysis in the Fuzzy51 package.

The "fuzzy knowledge base" block is used to choose appropriate bases for a particular circumstance and deliver them to the "fuzzy controller" block. Figure 2.4 depicts this. A computer system's knowledge base holds sophisticated information.

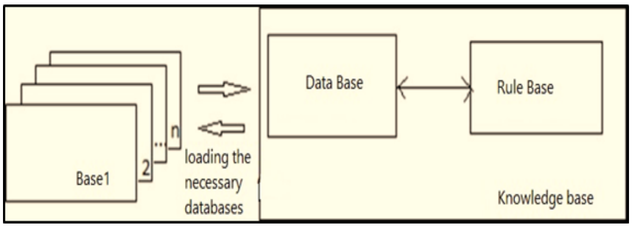


Fig. 2.4. Block "Knowledge Base".

Concerning the universality of fuzzy knowledge bases (FKBs) or a group of FKBs, it is clear that they are intended to handle unique object management challenges. For example, FKB No. 1 may do very well in addressing a certain issue, resulting in good outcomes. However, when confronted with additional issues, the results may be less than ideal. This means that each FKB is adapted to a particular set of circumstances.

Alternatively, a large knowledge base is required when the control object is vast, needing various circumstances or dynamic changes. A collection of FKBs is necessary to adequately handle all object management circumstances. The difficulty, however, is that a single FKB cannot contain all conceivable situations since it would hamper problem-solving efficiency, ease, and speed.

As a result, this research adopts a strategy that entails combining numerous discrete little fuzzy knowledge bases, such as FKB1, FKB2, and so on, into a single fuzzy knowledge base. The FKB set is in charge of picking and loading the appropriate base or bases based on the system requirements. The purpose of this study is to investigate ways to improve the system's flexibility and speed by introducing modifications or switches in the FKB.

Moving on to the "Control object" block, Figure 2.5 depicts the control object's basic structure. It is critical to emphasise the components that receive control signals (input points) and those that alter the environment (exit points). It is worth mentioning that the control object's particular structure might vary [4].

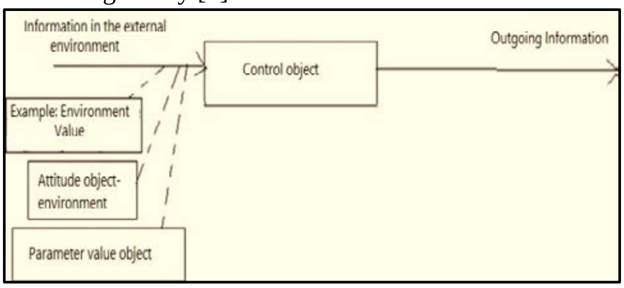


Fig 2.5. Description of the simple structure of the control object.

A fuzzy controller (FC) is the central component of such systems. NK may work in conjunction with a microcontroller or by alone.

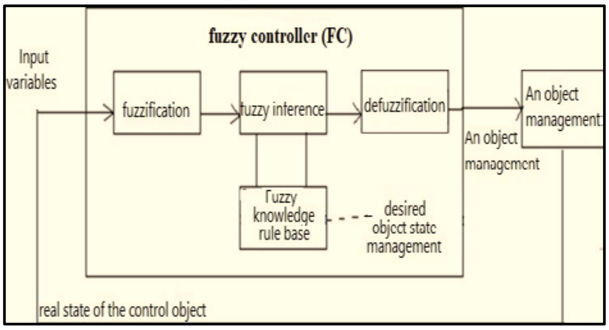


Fig.2.6. Structure of a system with fuzzy control logic

It is critical to establish the major parameters that will assist following operations while doing system analysis. These activities include investigating the controller's fuzzy structure (the amount of input and output variables), identifying language variables, and developing the rules regulating the use of the rule base to properly regulate the system.

Once the essential tasks at this stage are established, the associated rule base of the form "if..., then...", as well as the applicable fuzzy knowledge base (FKB) for the specific job, are implemented. However, since the knowledge base is not universally applicable in all cases, it may be required to choose one or more unique knowledge bases to improve the system's ease and efficiency. The fuzzy controller, which is the most important component of the system, processes the input variables from the control block and the related FKB for the selected object.

The control object's input variables are fuzzified, changing them into fuzzy form. These fuzzy variables are then subjected to fuzzy logic operations before being defuzzified, producing crisp values to control the object. In the context of robot control, for example, the object's control inputs may comprise parameters such as the form of the environment in which the robot works, the existence or absence of obstacles, the manoeuvrability of impediments, the robot's maximum rotation angle, and its speed [5].

Following the incorporation of these data into the fuzzy controller, rules for the robot's movement (e.g., speed, direction) are formed. It is critical to develop these guidelines prior to defuzzification. The robot now has explicit movement instructions after defuzzification.

	X1	X2	X3	X4	X5	Y1	Y2	Y3	Y4	Y5
1 >	in [050-150]									
2	<050									
3	>150									

Fig. 2.7: Manipulator of fuzzy control tactics.

Figure 2.7 illustrates the "Fuzzy Control Tactics Manipulator" program's interface. This programme makes it easier to transition between fuzzy knowledge bases for various activities. It is structured as a collection of subsystems that serve several phases, such as "Strategy settings" and "Data interchange."

B. Strategy settings tab (1).

The Tactics Selection Table group displays the rules in a tabular manner that may hold up to 255 rows. Each rule is structured as "IIF X1∈... and X2∈ ... and X3∈ ... and X4 ∈ ... and X5 ∈ ... and Y1 ∈ ... and Y2 ∈ ... and Y3 ∈ ... and Y4∈ ... and Y5∈ ..." You may use the "Add new line" and "remove existing line" buttons to add a new line or remove the current line (marked with the symbol ">") in the rule serial number column.

You may provide the input linguistic variables and criteria for the current line in the tactics selection table using the "Row element" group. You may specify the input variable in the Variable block and the related condition in the Condition section [6].

For example, "inside" means being inside a certain range, "outside" denotes being outside the range, "=" signals equality, ">" denotes inequality, "" denotes less than, and ">" denotes more than. In the "Condition" box, insert the values for the range's start point, end point, and comparison points. After finishing the task, you may proceed by selecting the "Add Item" button.

You may assign a strategy and a remark to the "Description of tactics" group. You may add a remark in the saved file if you use the "Select source file" button. Finally, when you've completed all of the required processes, click the "Assign tactics" button. To save and open the strategy, use the "Save strategy" and "Open strategy" buttons, respectively.

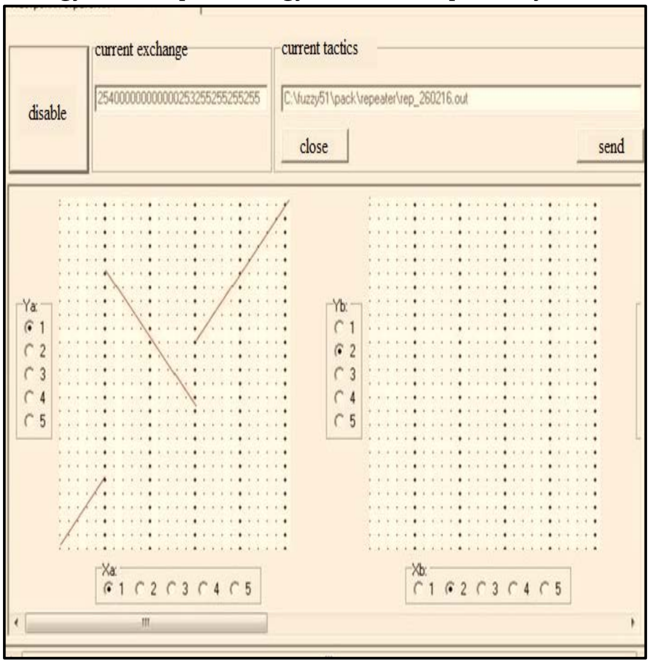


Fig. 2.8: Manipulator of fuzzy control tactics. Data exchange tab.

The "Data Exchange" page allows you to see up to 5 phase trajectories created by the constructed fuzzy controller, as well as their respective input and output variables (X and Y, numbered from 1 to 5). By pressing the "Enable" button, the trajectory and its values appear in the "Current Tactics" and "Current Exchange" sections. You may define the techniques used in the "Current techniques" group. The first 15 characters in the Current Tactics group are reserved for 5-byte input variables (X1 to X5), while the final 15 characters are reserved for 5-byte output variables (Y1 to Y5) (see Figure 2.9).

Fig. 2.9. Block "Current exchange" and "Current tactics".

To address these challenges, the VR simulation learning environment combines a mobile robot control Fuzzy Controller (FC). A mix of Fuzzy51 and VR programmes is used to support the laboratory workshop. Figure 2.10 depicts the organisational structure and the interplay of various components.

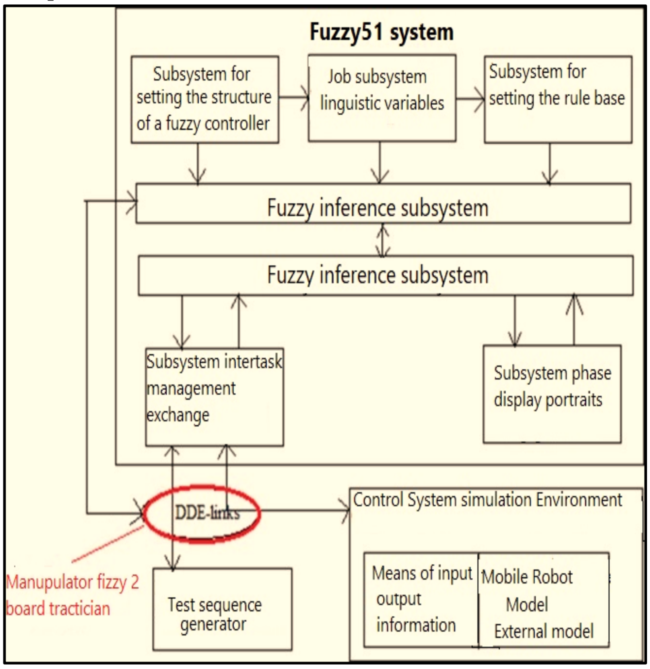
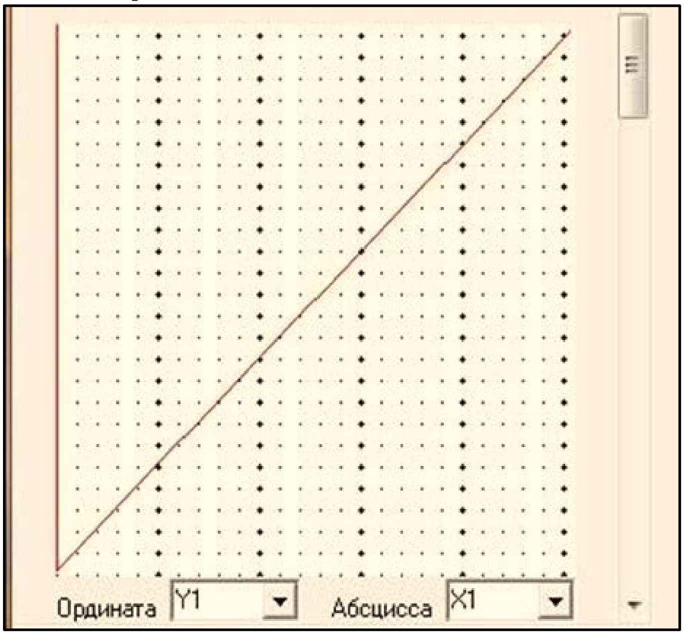


Fig. 2.10. Generalized structure of the software package.

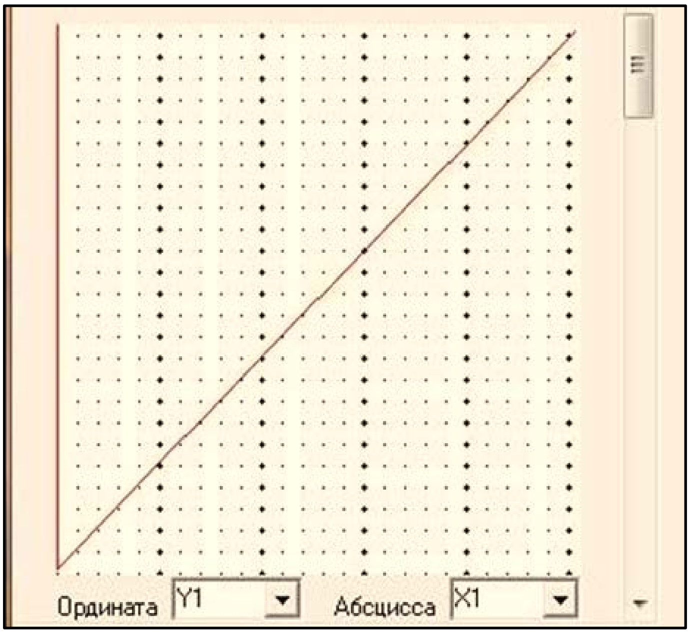
The VR simulation environment integrates the usage of Fuzzy51 through DDE (Dynamic Data Exchange), allowing the assessment of the correctness of the built Neurocontroller (NC) by evaluating decision accuracy in any given state of the environmental model. Our programme, "Fuzzy Control Tactics Manipulator," works inside the DDE_links architecture. To begin working in Fuzzy51, launch the NDT project and change the study mode to "External (DDE)" under the "Research" tab [7].

The exchange procedure begins by pushing the "Start computation" button on the button-panel. The "Fuzzy Control Tactics Manipulator" programme, which can be found under the "Strategy Settings" page, is used to keep input variables and tactics up to date. The "Enable" button in the "Data sharing" tab enables data sharing. The Fuzzy51 package's "Test" function is used to test the generated transducer by hitting the "Exchange inputs" button.

Returning to the Fuzzy Control Tactics Manipulator programme, selecting the Data Exchange option, and examining the findings is the last stage. Consider the following example, which involves toggling between two functions: the inverter function and the repeater function. To begin, open the file and inspect the output of the inverter function in the Fuzzy51 package, which is specified as Y=not(X). View the repeater function file and result in the Fuzzy51 package (refer to Figure 2.11), where the repeater function is specified as Y=X .



(a)



(b)

Fig. 2.11. (a,b) Repeater function in Fuzzy 51 package

The knowledge basis of these two functions is then exchanged. For example, with an input value of X ranging from 0 to 50, the value of the inverter function (Y=not(X)) is produced, and with X=50, the value of the repeater function (Y=X) is acquired. Figure 2.12 (bold red lines) depicts the theoretical result:

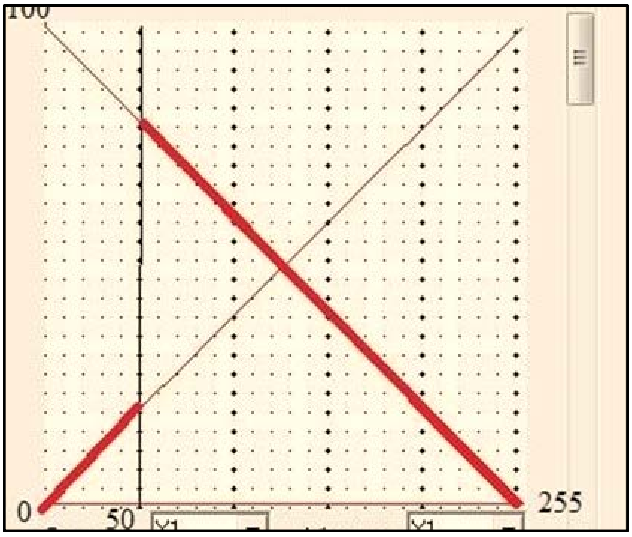


Fig. 2.12: Theoretical result in case of base 2 switching of inverter and repeater functions before and after X=50.

In this scenario, the kind of resultant structure in the modelling system (as indicated in Fig. 2.1) corresponds to Fig. 2.13.

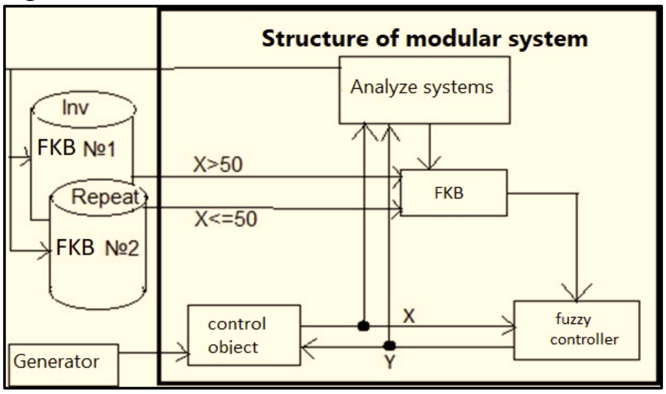
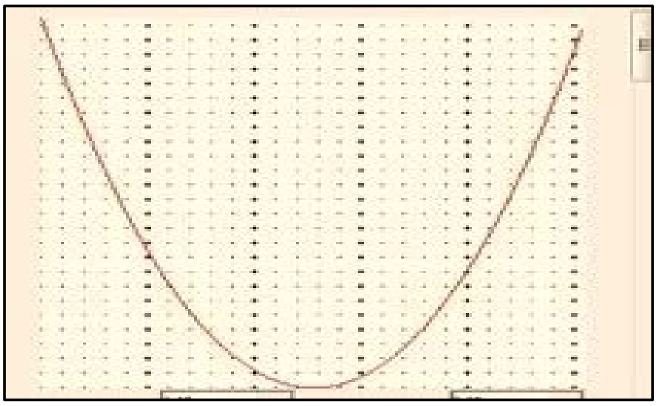
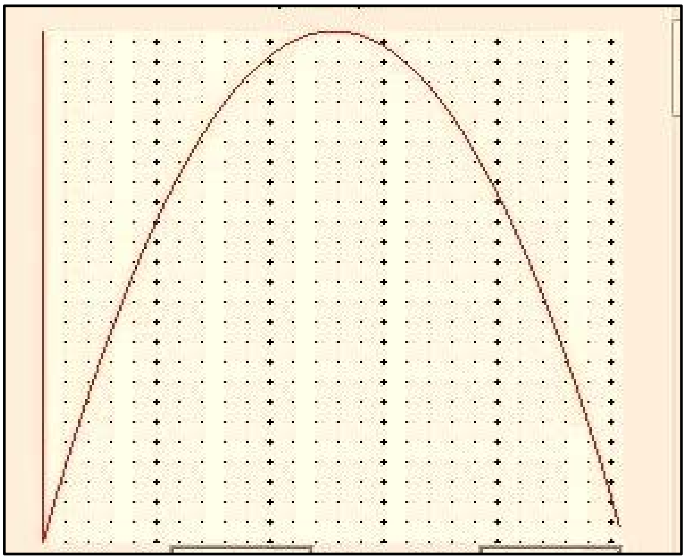


Fig. 2.13. Switching simulation system between two functions (inverter and repeater).

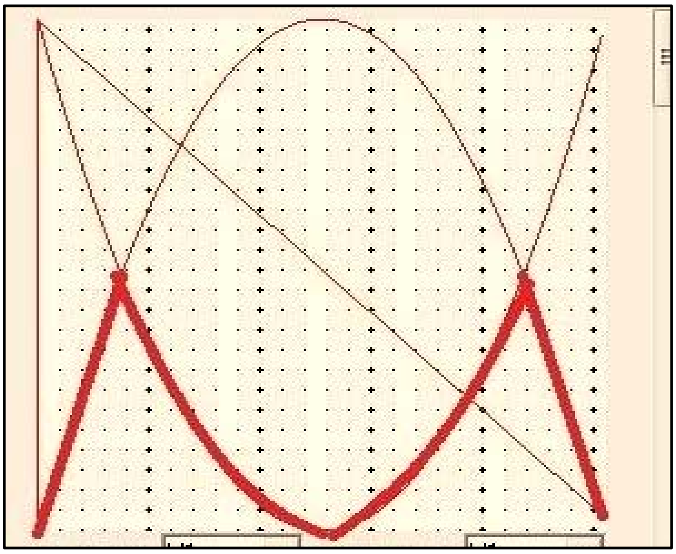
In the Fuzzy Control Tactics Manipulator programme, consider switching between two functions of the kind y=x2 and y= not(x2). Figure 2.14, c depicts the theoretical conclusion [8].



(a)



(b)



(c)

Fig. 2.14. Graph of 2 functions of the form y=$x2$(fig. a) and y= not($x2$).(fig. b) and their switching (fig. c, thick red lines).

In figure 2.15 depicts the simulation setup and experimental results. Where X=37 and X=127 are the switching points for these two functions [9].

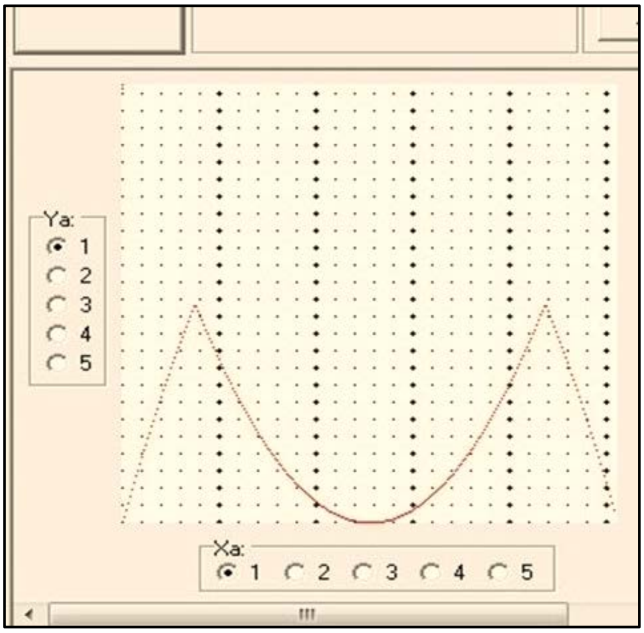


Fig. 2.15. Description of input variables X1 and tactics.

III. KEY FINDINGS

When studying control algorithms for moving objects in an environment with fixed obstacles, it is assumed to develop a fuzzy controller with the structure shown in Fig. 3.1.

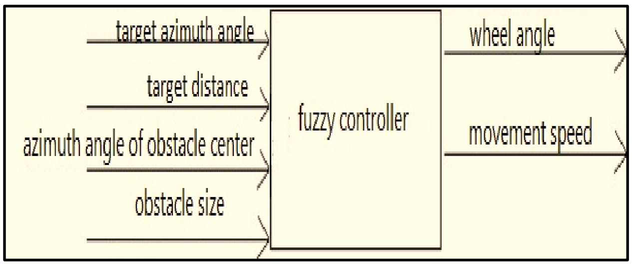


Fig. 3.1. The structure of a fuzzy controller in a system with VS without memory.

All inputs and outputs of the fuzzy controller are presented in a byte grid (value range: 0 - 255). In order to provide the required range of values for the input variables "target azimuth angle", “target distance”, “obstacle center azimuth angle”, “obstacle size” and output variables “wheel steering angle”, “moving speed”.

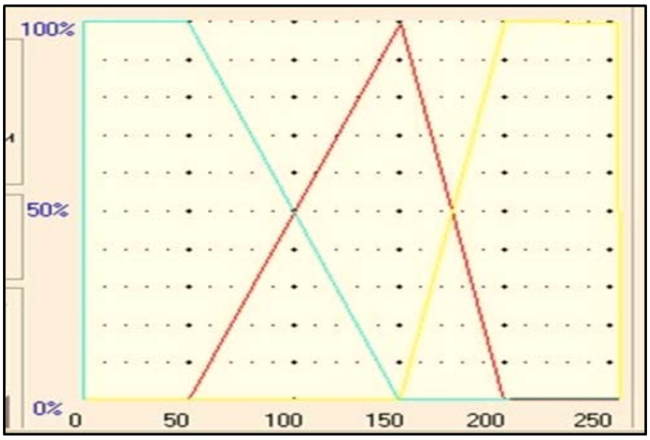


Fig.3.2. Terms of input variables X1 (angle per target)

Figure 3.2 depicts the linguistic input variable X1, which represents the angle to the target. Its meanings are stated in terms of "left, centre, and right". To physically implement a linguistic variable, it is important to determine the specific physical values that correspond to the variable's terms. Allow the variable X1 to take any value between zero and 255 in this scenario. Each value inside this given range can be associated with a numerical value between zero and one, according to fuzzy set theory. This numerical value shows the degree to which the specified physical distance is associated with one or more terms of the linguistic variable X1.

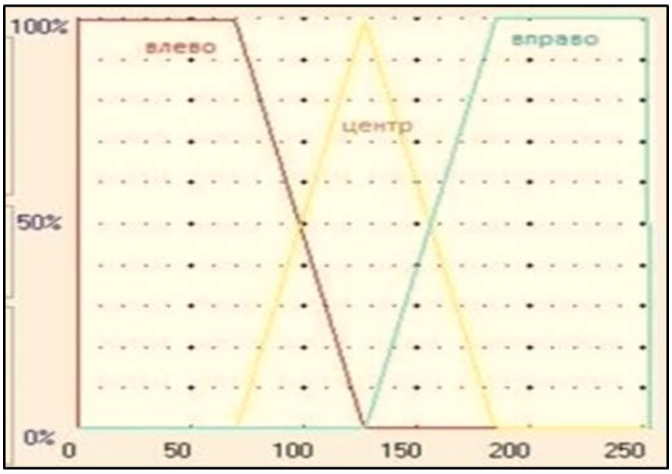


Fig.3.3: Output linguistic variables

a) "Left," "centre," and "right" serve as the semantic definitions for the angle-representing linguistic input variable X1, which represents the direction to the destination. The physical values assigned to these terms range from 0 to 255.

b) To achieve exact stopping on the target, the linguistic output variables U1, representing the angle of rotation (left, centre, and right), and Y2, representing the robot's speed, are chosen. The speed variable has three values: "low," "medium," and "high." The following outcomes were attained throughout the study [10]:

1. The creation of a system framework for modelling the control of a mobile robot, with the capability of switching between knowledge bases.
2. A comparison study was performed, comparing the theoretical and empirical data. experimental methods for dynamically changing fuzzy knowledge bases, resulting in consistent findings.
3. The research explains how to build a simulation programmed, including how to transition between several fuzzy knowledge bases. Furthermore, the control of a mobile robot is examined and analyzed to address control challenges and ensure desirable trajectory movements.

IV. CONCLUSION

This article is concerned with control methods for moving things around stationary barriers. The research focuses on the creation of a specific framework for a fuzzy controller. A byte grid represents input and output variables such as target azimuth angle, target distance, obstacle centre azimuth angle, obstacle size, wheel steering angle, and movement speed. The physical values of the linguistic variables X1, U1, and Y2 are specified. The research effectively creates a framework for modelling mobile robot control that can transition across knowledge bases. A comparison of theoretical and experimental techniques to dynamically altering fuzzy knowledge bases is offered, as well as processes for developing a simulation program.

ACKNOWLEDGMENT

The present study received financial assistance from the Institute of Information Technologies at the Russian Technological University RTU MIREA, located in Moscow, Russia, as well as from Mustansiriyah University in Iraq.

REFERENCES

[1] T. Blender, T. Buchner, B. Fernandez, B. Pichlmaier and C. Schlegel, "Managing a Mobile Agricultural Robot Swarm for a seeding task," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, 2016, pp. 6879-6886. doi: 10.1109/IECON.2016.7793638.

[2] K. Hernandez, B. Bacca and B. Posso, "Multi-goal Path Planning Autonomous System for Picking up and Delivery Tasks in Mobile

[3] Robotics," in IEEE Latin America Transactions, vol. 15, no. 2, pp. 232-238, Feb. 2017. doi: 10.1109/TLA.2017.7854617.

[4] M. Langerwisch and B. Wagner, "Dynamic path planning for coordinated motion of multiple mobile robots," 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, 2011, pp. 1989-1994. doi: 10.1109/ITSC.2011.6083047.

[5] S. Liu, D. Sun and C. Zhu, "Coordinated Motion Planning for Multiple Mobile Robots Along Designed Paths With Formation Requirement," in IEEE/ASME Transactions on Mechatronics, vol. 16, no. 6, pp. 1021-1031, Dec. 2011. doi: 10.1109/TMECH.2010.2070843.

[6] Noritaka Sato, Kazuyuki Kon, Hiroaki Fukushima and Fumitoshi Matsuno, "Mapbased Navigation Interfacefor Multiple Rescue Robots," in IEEE International Workshop on Safety, Security and Rescue Robotics Sendai, Japan, October 2008

[7] M. Tazir, O. Azouaoui, M. Hazerchi and M. Brahimi, "Mobile robot path planning for complex dynamic environments, " in International Conference on Advanced Robotics (ICAR), 2015

[8] Vojtech Vonásek, Martin Saska, Karel Kosnar and Libor Preucil, "Global motion planning for modular robots with local motion primitives," in IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe, Germany, May 6-10, 2013.

[9] I.A. Vasiliev, S.A. Polovko, E.Yu. Smirnova St. Petersburg, Russia. Organization of group control of mobile robots for tasks of special robotics.

[10] A.I. Diveev, E.Yu. Shmalko. Synthesis of control for an autonomous group of robots with phase constraints by the method of a multilayer network operator with prioritization.

Al-khafaji Israa M. Abdalameer — Postgraduate student at Institute of Information Technologies. Russian Technological University RTU MIREA. Assistant teacher at Mustansiriyah University, Collage of Science, department of Computer Science, Iraq. Research interests: information technology, deep learning, machine learning and software development. misnew6@gmail.com, ism@uomustansiriyah.edu.iq, 119454, Central Federal District, Vernadskogo Avenue, 78, Russia; phone no. +7(985)177-93-39.