A NOVEL PROTOTYPE MODEL FOR SWARM MOBILE ROBOT NAVIGATION BASED FUZZY LOGIC CONTROLLER

Sherif Kamel Hussein[1]and Mashael Amer Al-Mutairi[2]

[1]Associate Professor – Department of Communications and Computer Engineering
October University for Modern Sciences and Arts ,Giza- Egypt
[1]Head of Computer Science Department, Arab East Colleges for Graduate Studies,
Riyadh, KSA
[2]Co- Author: Master of Computer Science,Arab East Colleges for Graduate Studies,-
Riyadh, KSA

ABSTRACT

Autonomous mobile robots have been used to carry out different tasks without continuous human guidance. To achieve the tasks, they must be able to navigate and avoid different kinds of obstacles that faced them. Navigation means that the robot can move through the environment to reach a destination. Obstacles avoidance considers a challenge which robot must overcome. In this work, the authors propose an efficient technique for obstacles avoidance through navigation of swarm mobile robot in an unstructured environment. All robots cooperate with each other to avoid obstacles. The robots detect the obstacles position around them and store their positions in shared memory. By accessing the shared memory, the other robots of the swarm can avoid the detected obstacles when they face them. To implement this idea, the Authors used a MATLAB® and V-REP® (Virtual Robot Experimentation Platform).

KEYWORDS

mobile robot, swarm robot, navigation, obstacle avoidance, fuzzy logic controller

1. INTRODUCTION

The field of robotics is a very interesting and useful. The term “*Robot*” is derived from the Czech word “Robota” which means menial laborer. It has appeared on January 25, 1921 [1]. Robots can be used in manufacturing environments, in underwater and space exploration, for helping the disabled, or even for entertainments. There are different kinds of robot: manual handling device, fixed sequence robot, variable sequence robot, playback robot, numerical control robot, and intelligent robot [2].

"An intelligent robot is a mechanical creature which can function autonomously". The intelligent robot has to be able to handle ambiguous, conflicting, cluttering and uncertain situations. It can cope with all vagaries of the real world and learn from its interaction with the environment [3]. Many people thought that the robot must be anthropomorphic (human-like). In the real world, the robots have a variety of shapes depending on their jobs [1]. Drive systems for robots are diverse depending on the environment where the robot exists. There is a mobile robot with wheels, legs or continuous track. Another kind is aerial robots (**UAV** Unmanned Aerial Vehicle) which can fly using propellers. The aquatic is the third one (**AUV** Autonomous Underwater Vehicle). This robot can move under or over water [4].

DOI: 10.5121/ijcsit.2019.11206

1.1 Mobile Robot

Although that the fascinating successes of robots is in industrial manufacturing area by using robot arms. Nevertheless, this kind of robot has a limited ability in mobility which is required to achieve many essential tasks [5]. Mobile robot is the kind of robot that uses wheels, legs or both to move across the ground [6]. In fact, the mobile robots differ in how they move. There are tracked, wheeled, legged, leg-wheeled, segmented, climbing and hopping. Spherical robot "ball-like" is also considered as a mobile robot even it does not have wheels or legs. Legged robot may be used single or multiple legs such as two legs (biped), four legs and six legs. Each one provides specific advantages and has different uses. The wheeled robot is a most popular mobile robotics that comes with two, three or more wheels. It is an absolutely practical type that can be used in many vital areas in our life.

Navigation means the robot moves from an initial position to reach a destination. The autonomous mobile robot is the age-old dream that people have tried to achieve it for centuries. Autonomous mobile robot navigation can be described by these three basic questions: Where is robot? Where is robot going? How does robot get there? [7]. The Navigation system can be dividedinto various subtasks. Identifying the present location of the robot and things around it, which is called "*Localization*". Avoiding any collisions. Defining a path to reach the goal. Make sure there is no conflict between the last two subtasks [8]. In the "*PathPlanning"*, the robot attempts to find an optimal path to reach a prescribed fixed goal. The obstacles in the environment can be stationary or moving (like people walking around the robot) and there are several methods to avoid each type.

The environment in which the robot can navigate can be classified as: Structured (Known) Environments, Partially Structured Environments, and Unstructured (Unknown) Environments.Based on the environment, the navigation is dividedinto two categories: *Indoor* and *Outdoor*. The autonomous mobile robots that acquire and maintain models of the outside environment when it is the outdoor type, while the indoor robots acquire the inside environment models. In some cases, "*Landmarks*" such as beacons, visual patterns or induction loops are used to help robot for navigation. They can be geometric shapes such as circles or lines and must be easy to identify. The landmarks have known position, relative to which a robot can localize itself. [9].

1.2 Swarm Robot

"*Swarm robot*" consists of a number of robots that deal with tasks which are impossible to be achieved by an individual robot [10]. Another definition of swarm robot is "a novel approach to the coordination of large numbers of robots" [11]. In 2004, Gerardo Beni [10] was the first one who used "*Swarm*" term to indicate collective robots. Distributed robotics, collective robotics, and robot colonies are alternative terms [11].

In fact, swarm robot is not just a "group". It has significant features which are inspired by different magnificent natural swarms like ant colonies, the swarm of bees, bird flocking, animal herding, the colony of bacteria and fish schooling. The most considerable features are:

1. The swarm consists of (quasi) identical individuals.
2. The interactions between members are based on easy rules that use only local data.
3. Decentralized control.
4. Connections among the members happen in a localized approach.
5. The comprehensive behavior of the system results from the interactions of singular members with each other and with the environment around them.

6. The response of the system is completely strong and adaptive with several changes in the environment [12].

The swarm robotics has a set of fundamental problems that must address with them. These problemsare divided into three main groups. The first group is pattern formation problems like how to deploy a group of distributed antennas of sensors, robots aggregation, how to construct a map for an environment and how to create gradients and so on. The second group is the problems relate to some of the environmental factors for example: how to find the source of a chemical plume, how to search for the target, foraging, homing, etc. The last problems group is that focused on the sophisticated group behavior like mining, flocking, collaborative transport and shepherding [10].

Swarm robotics is an optimal choice for performing tasks that require parallelism and redundancy. Swarm mobile robot applications become a very interesting and promising field. In recent time, a variety of applications appear in our life like items transportation, military operations, and demining or monitoring. To achieve these diverse tasks, a swarm mobile robot needs to navigate in the environment. Through navigation, the robots will face many obstacles and they need a technique to avoid them.

Fuzzy Logic, Genetic Algorithm, Particle Swarm Optimization (PSO), and Neural Network (NN) are examples of techniques the robots use to avoid different kinds of obstacles around them. Instead of these techniques, some robots help each other to recognize the obstacles and precisely detect their locations. They exchange messages contain navigation information to guide each other's navigation.

In this paper, the authors consider a situation where a swarm of robots moving at the same time. The robots are deployed in unknown area. Each individual robot has sensors that allow it to discover the environment and detect exist obstacles. When robot determines obstacle location, it sends the detected location to memory. The memory can be accessed by other robots in the swarm to use stored information. In this proposed system, the robots will help each other by sharing obstacle locations.

2. LITERATURE REVIEW

Navigation of mobile robots is a big challenge. There are many algorithms used for this purpose. In [13] the researchers presented a new Fuzzy Logic Controller (FLC) for wheeled mobile robots. They used that Fuzzy Logic Controller (FLC) two times. The first time with five membership functions and the second one with seven membership functions. The results showed that the first situation gives better performance. Optimal fuzzy logic controller is proposed in [14]. To optimize the outcomes of a Sugeno fuzzy logic optimal controller the researchers used Genetic algorithms and gradient method. The authors in [15] proposed an on-line navigation technique in an unstructured environment with movable obstacles using Fuzzy logic techniques. Narvydas, Simutis, and Raudonis proposed FUZZY logic, IF-THEN rules, and Genetic Algorithm for autonomous mobile robot control. Their experiment showed that the control using fuzzy logic gives best results than the other control which uses IF-THEN rules because there are no motors speed jumps [16].

Regarding the swarm mobile robot, many techniques are used for swarm robot navigation. Güzel et al proposed (APSO) ,a modified version of the conventional Particle Swarm Optimization (PSO) algorithm that allows swarm robot to overcome both static and moving obstacles and reach to the static and moving targets in a safe and collective manner [17]. A simple flocking task is presented in [18]. The researchers described a leaderless distributed flocking algorithm (LD) that uses the time derivative of the perceived center of the flock to align the robots without explicit knowledge of neighbor heading. The researchers used an embodied simulator and reinforcement learning techniques to optimize LD performance under different conditions. They showed that method can be used to improve performance and to gain insight into which algorithm components contribute most to system behavior. In article [19] the researchers used fuzzy logic to control the suction/thrust force between individuals in the swarm and Null-Space Behavioral control technique to apply control to the multitasking system of the swarm robot. They solved the problem with high flexibility by selecting the input/output, defuzzifier, and the fuzzy rules. In [20] a navigation system for robotic swarms is presented. The algorithm allows robots to exchange messages contain navigation information to guide each other's navigation. The authors used this algorithm in two scenarios. In the first scenario, the robots of a swarm guide a single robot to a destination. In the second scenario, the swarm robot navigates between two targets. Sugawara, Kazama, and Watanabe proposed a Virtual Dynamic Environment for Autonomous Robots (V-DEAR) that simulates chemical system such as pheromone. They deal indirect communication between the robots and investigate a foraging behavior by multi-robot system [21]. In [22] the authors presented an effective algorithm that deploys a network of radio beacons which help the robot in coverage and navigation. The presented approach is extendable to multi-robot applications, in which the network can be thought of as a multi-purpose task manager.

3. THE NEWLY PROPOSED SYSTEM – SWARM MOBILE ROBOT :

This paper presents a simple and efficient system to help a swarm mobile robot to avoid obstacles. The main contribution of this work is to solve the obstacles avoidance problem by allowing the robots to cooperate with each other to determine the obstacles. Each robot uses the sensory information to detect the location of any obstacle around it. By using the wireless technique, the robot will store the obstacles information in a shared memory. All robots can access the information stored in this memory to avoid any obstacle that will face them. This system will help the individual robot to benefit from other robot effort. It does not need to detect all obstacles by itself

3.1 FUNCTIONAL REQUIREMENTS

The system allows robots to:

- Move through the environment.
- Avoid obstacles.
- Add new obstacle location in the shared memory.
- Access the shared memory to check if there is an obstacle in its way.

3.2 NON FUNCTIONAL REQUIREMENTS

- **Implementation constraint:**
  This system simulates only Khepera III robots by using a MATLAB® and V-REP® simulator.
- **Easy to use requirements**
  This system is simple and it shows messages that describe the state of the robot when it changes. These messages allow the user to understand what happens.

3.3 DOMAIN CLASS DIAGRAM

In the domain class diagram there are four classes: Area, Obstacle, Robot, and Target. The classes' attributes and relations between classes are shown in Figure 1.

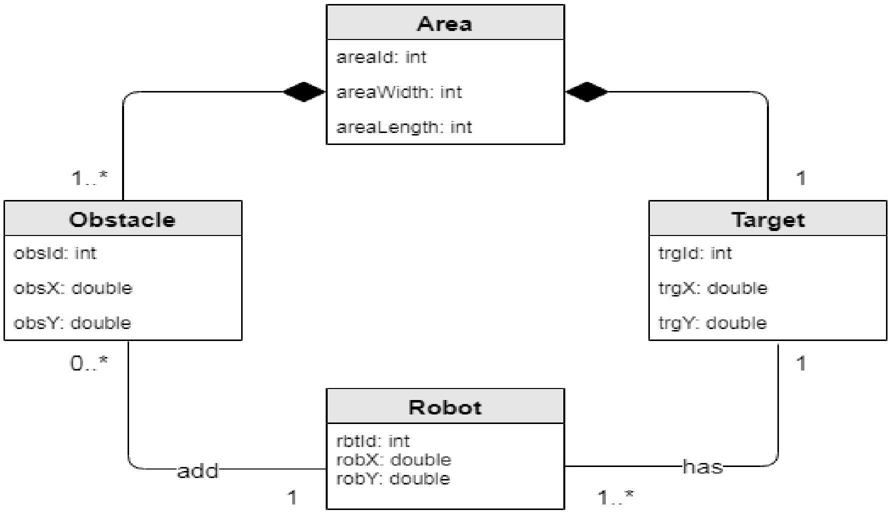


Figure.1 Domain class diagram

3.4 State Machine Diagrams

Figure 2 and Figure 3 provide the state of robot and obstacle respectively. Each robot is active until it reaches the required destination.Its state will be changed and it becomes ideal. The obstacle can be either detected or not detected by the robot.

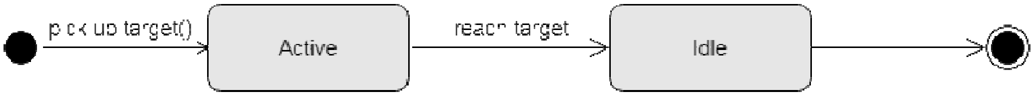


Figure 2 State machine diagram for Robot

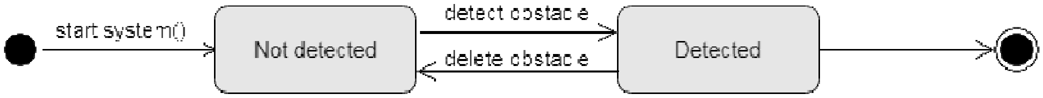


Figure 3State machine diagram for Obstacle

3.5 Sequence Diagram

Figure 4. Illustrates clear information about the sequence of the steps that the robot follows to arrive the desired target.

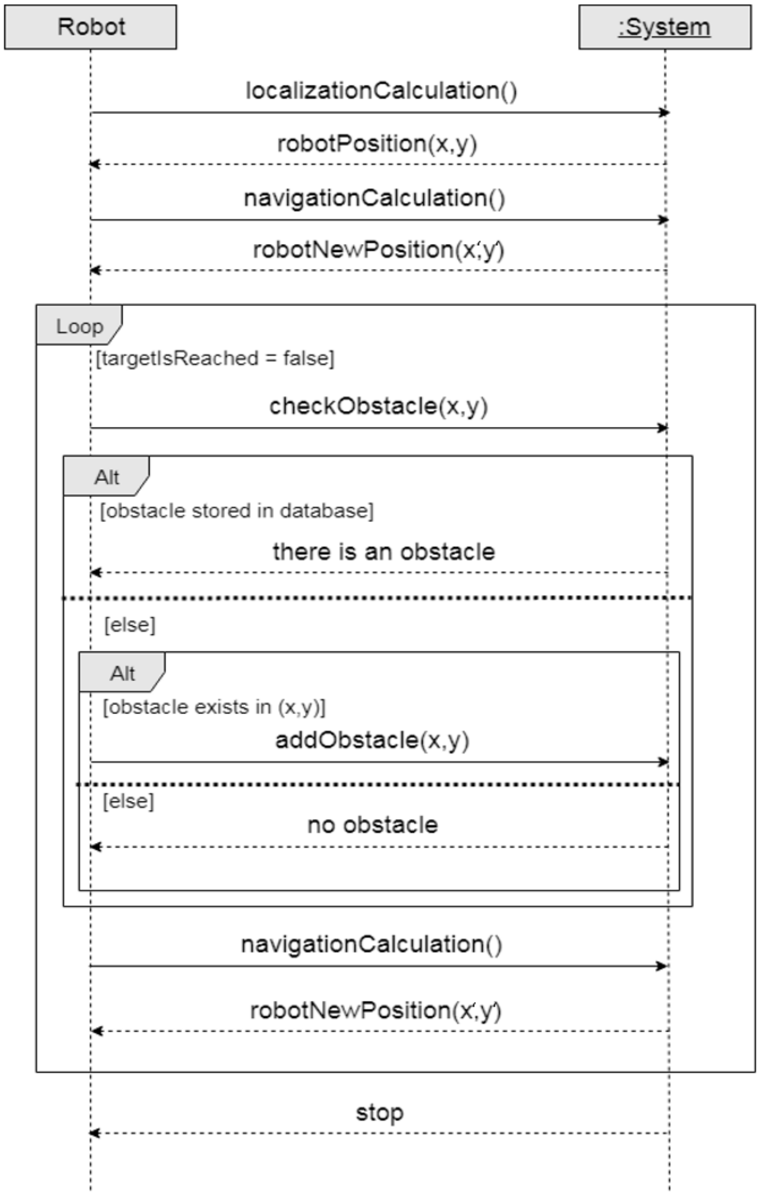


Figure 4.Sequence Diagram

4. SYSTEM IMPLEMENTATION

The implementation is based on using the Fuzzy Logic Controllers to control both the Navigation and the obstacles avoidance systems.

4.1 SYSTEM TOOLS

4.1.1 MATLAB®

Matlab is a distinct programming language with its own advanced data structures. Its basic data element is an array. Matlab includes built-in editing and debugging tools and supports object-oriented programming model. For these features, Matlab becomes a prime tool for teaching and researching [23]. In addition, Matlab supports various programming languages such as C, C++, Fortran, Java, and Python.

Matlab provides wide and sophisticated toolboxes for a variety of scientific and engineering applications.Some of these toolboxes are free included with Matlab, the others you must buy them. You can create your own toolbox and share it with other Matlab users. The toolboxes are classified as follow: parallel computing, math, statistics, and optimization, control systems, signal processing and wireless communications, image processing and computer vision, test and measurement, computational finance, computational biology, code generation, application deployment and database access and reporting .

In the proposed system,Fuzzy Logic Toolbox is used to design, simulate and analyze systems based on fuzzy logic. It provides a lot of functions and apps. You can use this toolbox to easily design a fuzzy inference system. It supports both Standard Mamdani and Sugeno-fuzzy inference systems. The toolbox provides eleven kinds of built-in membership functions which built from four basic functions: Piece-wise linear functions, Gaussian distribution function, Sigmoid curve, and Quadratic and cubic polynomial curves. It also enables you to create your own membership function. The toolbox has tools for modeling system behavior as simple (AND, OR, and NOT) logic rules and then implement them in a fuzzy inference system

Matlab version 9.4 is used in the design of the Swarm Mobile Robot sytem. This release comes with several new features and enhancements in many sections such as the live editor, mathematics, graphics, data import and export, data analysis, app building, and hardware support. Its performance is improved as well. The Matlab startup time is enhanced. Also, increasing the performance of the execution engine for popular programming patterns [24].

4.1.1 V-REP®

V-REP is a powerful and flexible 3D robot simulator that simulates different kinds of mobile and non-mobile robots. V-REP stands for'Virtual Robot Experimentation Platform' developed by Coppelia Robotics. It comes with a suitable GUI which facilitates the basic functions. It supports Linux, macOS, and Windows platforms. In V-REP, the scene, model and the simulator itself can be controlled in many easy ways via an embedded script written in Lua language, a remote API client, a plugin and so on. V-REP is a powerful simulator for multi-robot systems. V-REP has a remote API used for controlling a simulation from an external application. This function is available for six programming languages Matlab, Python, Java, C/C++, Lua, and Octave. V-REP is a perfect tool that can be used for many purposes, for example, robotics,fast prototyping, factory automation simulations, developing algorithms and remote monitoring. V-REP offers many features like collision detection, minimum distance calculation, inverse /forward kinematics, proximity sensor simulation, vision sensor simulation, and path /motion planning [25].In the proposed system,V-REP 3.5.0 version (free educational license) is used.

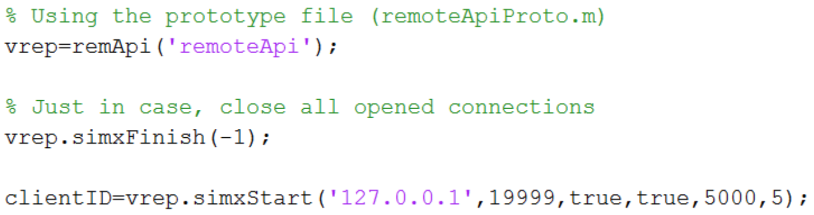
Khepera is a common robot manufactured by K-team in 1991. It is considered a miniature robot with about 130 mm diameter.Because of its small size, the robot was known as “cafard” which means “cockroach” in the French language. Prof. Nicoud, changed this name and called it “Khepera”, which is an Egyptian god having the head of a cockroach [26].KheperaIII has eight infrared proximity/light sensors with up to 30cm range and five ultrasonic sensors with range 20cm to 4 meters. For line following applications, It has two infra-red ground proximity sensors. Many extension modules are available for this robot such as camera, radio communication, and gripper. Khepera III robot can be controlled by the computer via serial port [27].

4.2 Project Techniques

1. Build the environment and develop the obstacles and set the location of each obstacle.
2. Implement navigation system based on the Fuzzy logic controller.
3. Design Fuzzy logic controller for obstacle avoidance by using the fuzzy logic toolbox and determine the input and output variables and set the rules.
4. Implement the swarm robot.
5. Implement the obstacle avoidance technique.

4.3 V-Rep Remote Api

In the proposed system, Matlab and a V-REP are used as remote API to control the robots. At firstto connect with V-REP, the following commands should be used:



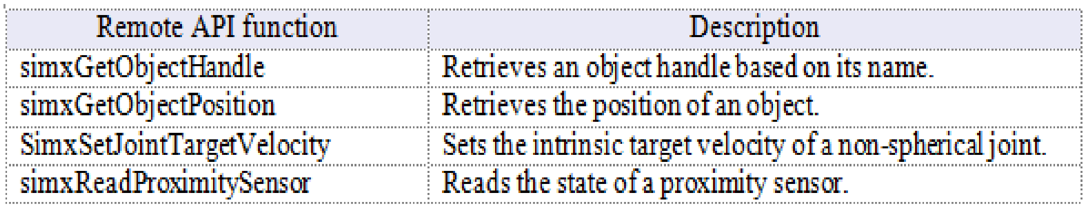
```
% Using the prototype file (remoteApiProto.m)
vrep=remApi('remoteApi');

% Just in case, close all opened connections
vrep.simxFinish(-1);

clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
```

The V-REP remote API contains many functions as shown in Table 1 that can be called by the Matlab.

Table 1 Remote API functions



Remote API function	Description
simxGetObjectHandle	Retrieves an object handle based on its name.
simxGetObjectPosition	Retrieves the position of an object.
SimxSetJointTargetVelocity	Sets the intrinsic target velocity of a non-spherical joint.
simxReadProximitySensor	Reads the state of a proximity sensor.

4.4 Navigation

The navigation system is very simple and effective. At the beginning, the speed is set for all robot. Then the distance between the robot and the target is calculated using Pythagoras theorem as follows:

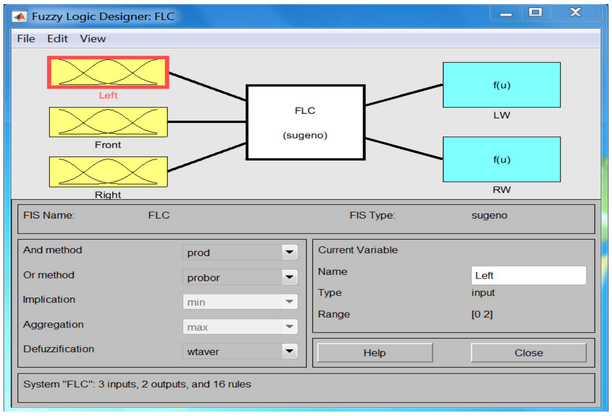
$$\text{Distance} = \left|\sqrt{(x\_robot - x\_target)^2 + (y\_robot - y\_target)^2}\right|$$

When a robot reaches the target (distance = 0), the speed is changed into zero then the robot will stop.

4.5 Obstacles Avoidance

Fuzzy Logic controller is created for the obstacles avoidance. The type of fuzzy inference system is Sugeno that is similar to the common method Mamdani. The main difference between them, that the Sugeno output can be either linear or constant.

As shown in Figure 5, the fuzzy logic controller has three input variables ‘left’, 'front' and ‘right. They indicate the values of the infrared proximity sensors of the robot. Each input variable has three trapezoidal membership functions (N Near, M Medium, and F Far). Figure 6 presents the input variables’ membership functions. There are two output variables LW and RW where LW is the velocity of the left wheel and RW is the velocity of the right. Each output variable has four constant membership functions (small = 1, medium = 2, high=3 and minus=-0.5). Output variables’ membership functions are shown in Figure 7.



FIS Name: FLC

FIS Type: sugeno

And method: prod

Or method: probor

Implication: min

Aggregation: max

Defuzzification: wtaver

Current Variable

Name: Left

Type: input

Range: [0 2]

System "FLC": 3 inputs, 2 outputs, and 16 rules

Figure 5 Fuzzy logic controller

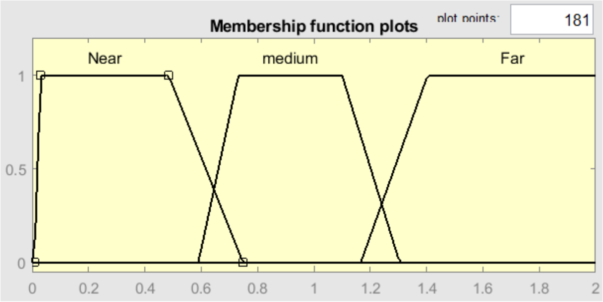


Figure 6 Left, front and right sensors’ membership functions

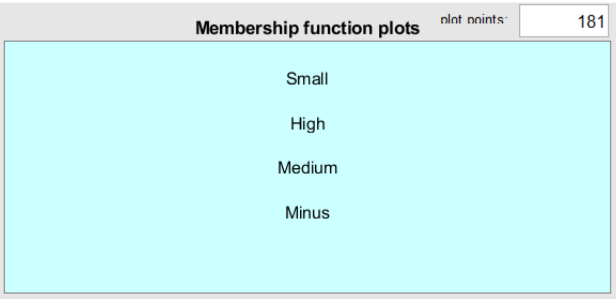
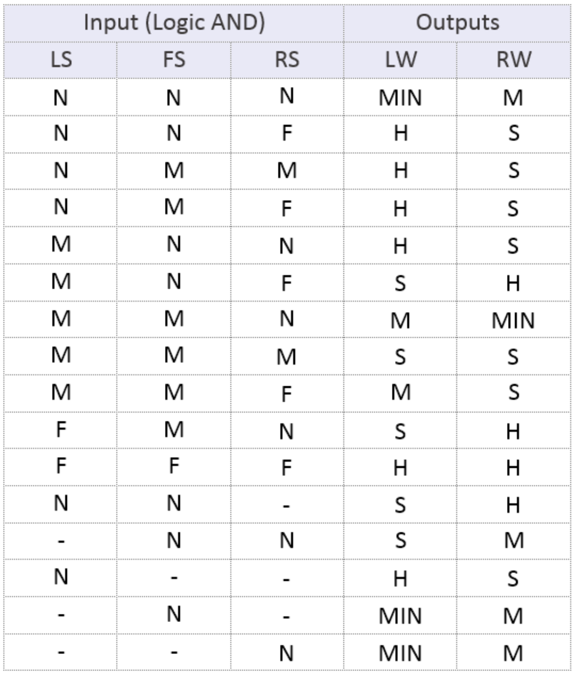


Figure 7 Left and right wheels’ speed membership functions

The fuzzy logic inference rules for the Obstacles Avoidance System are shown in Table2.

Table 2 Fuzzy logic inference rules



Input (Logic AND)			Outputs	
LS	FS	RS	LW	RW
N	N	N	MIN	M
N	N	F	H	S
N	M	M	H	S
N	M	F	H	S
M	N	N	H	S
M	N	F	S	H
M	M	N	M	MIN
M	M	M	S	S
M	M	F	M	S
F	M	N	S	H
F	F	F	H	H
N	N	-	S	H
-	N	N	S	M
N	-	-	H	S
-	N	-	MIN	M
-	-	N	MIN	M

5. SYSTEM FLOW CHART

In the proposed system,the simulated wheeled mobile robots will be organized in swarm. The robots navigate in an indoor and unstructured environment. In the environment, there are many stationary obstacles in unknown locations. All robots try to reach the same target. The target is also static and its location is known for each robot. The robots collaborate to avoid obstacles that will face them through navigation. The navigation system is autonomous and maples.

As shown in Figure 8 at the beginning, the robot localizes itself. It estimates its x-coordinate, y-coordinate by using sensory information. Then, the robot uses the navigation system to make its first step. After the first step, the robot checks if the target is reached?. If the answer is yes, the robot achieved its work and will stop. In the second case, the robot immediately accesses the shared memory to search for information about the next step.If the information exists. Two cases will face the robot. The first case, if the next step is free of an obstacle, the robot will continue navigation. Another case, if there is an obstacle towards robot. The robot will avoid this obstacle by changing its direction. Then, the robot will continue navigation.

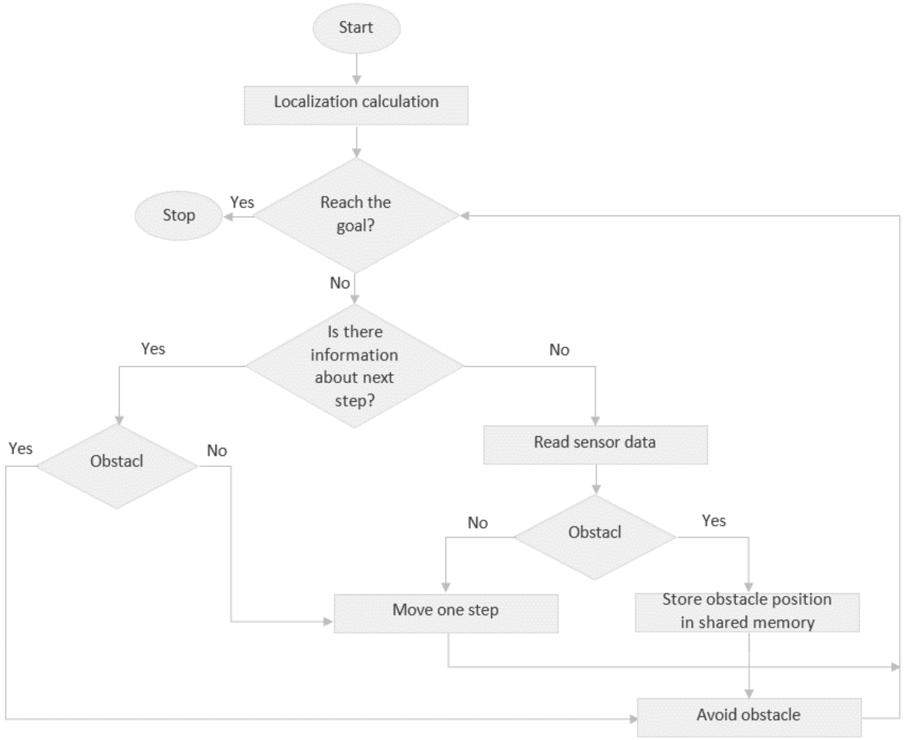


Figure 8 System flowchart

Sometimes, the robot will be the first one who traverses this location. Thus, there is no information in the shared memory that can guide the robot. The robot will be the initiator of this information. It will use its sensors to read information and check if the step has an obstacle or not. When the robot detects an obstacle. It will change its direction to avoid it and store obstacle location (x-coordinate and y-coordinate) in the shared memory. If there is no obstacle in the specific step, the robot will also share this information by storing it in the memory. In both cases, after the storing step the robot will continue navigation. Other robots in the swarm can use stored information when they travel through this step.

Every time, the robot will check for the next step if it is the target or it was traversed before or it is a new. Then, it will decide what it must do. This cycle will repeat for each robot until it reaches the target.

6. Results And Analysis

6.1 Environment Without Obstacles

In this case, each robot moves toward the target directly. They do not need to use infrared sensors. The results show that the system is accurate and each robot stopped when reached the goal. Figures (9, 10, 11) illustrate the robots start locations, the robot's movement, and the robots when reached the goal respectively.

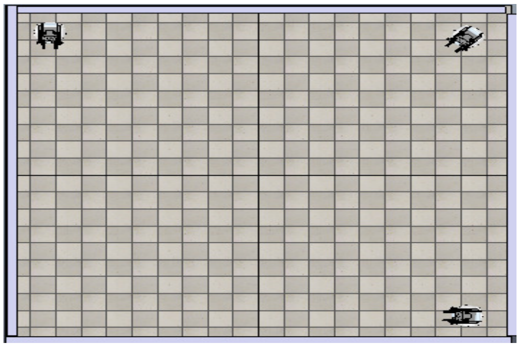


Figure 9 Navigation without obstacles (a)

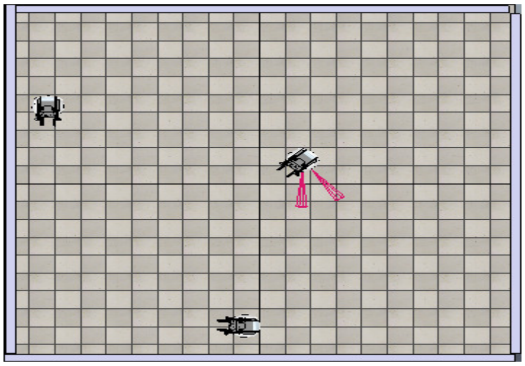


Figure 10 Navigation without obstacles (b)

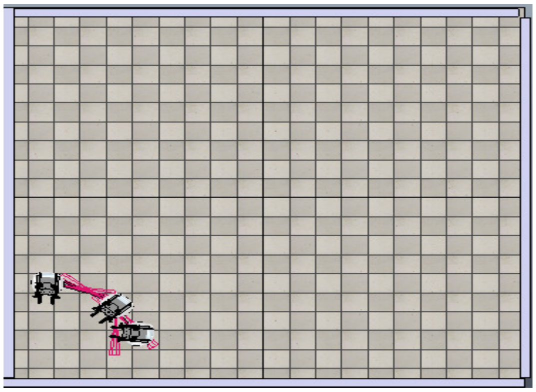


Figure 11 Navigation without obstacles (c)

6.2. ENVIRONMENT WITH OBSTACLES

Navigation with obstacles is more difficult than the first case. The robots will face obstacles in their way and they must detect them by using infrared proximity sensors and avoid them by using the Fuzzy Logic controller. Then, they store obstacles locations in the shared memory. The following figures (12, 13, 14) show swarm navigation in the environment with the obstacles.

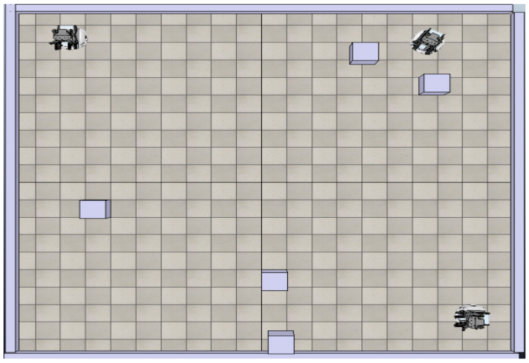


Figure 12 Navigation with obstacles (a)

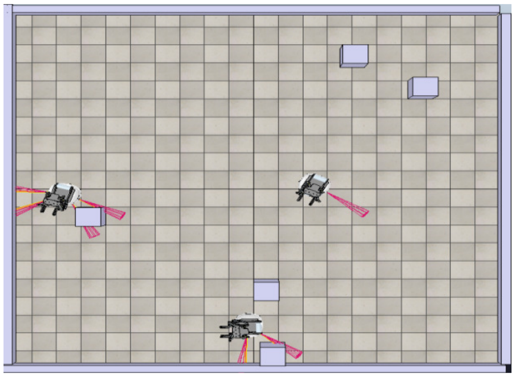


Figure 13 Navigation with obstacles (b)

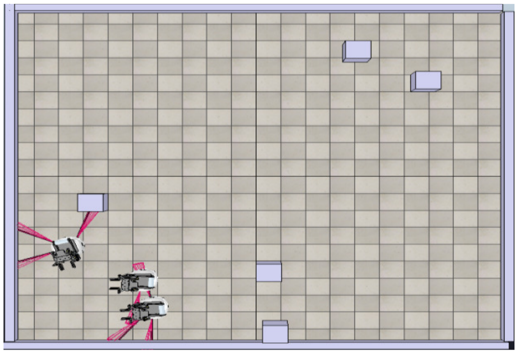


Figure 14 Navigation with obstacles (c)

7. Conclusion And Future Work

This system aims to implement a team of mobile robots that cooperate to detect the obstacles in an indoor environment. All robots go to the same destination which is a spatial location in the environment. The destination is static and it is known for all the robots. The environment has some of static obstacles that will be discovered by the swarm. During navigation, when a robot faced an obstacle, it will store its location in the memory and it can be accessed by all the robots. The experiments have been carried out in a simulated environment; the results indicated that the proposed system is simple and effective. The implementation is carried out using Matlab and V-REP as a remote API to control a simulated Khepera III robots from Matlab.

In the proposed system the angle between robot and target is between 0 to 90. In the future, the system will be enhanced to handle other angles cases. Also improve the obstacles avoidance controller to be more accurate. The proposed system worked only in a simple and static environment. The system will be enhanced to carry out other cases likedynamic goals and obstacles.

REFERENCES

[1] Murphy, R. 2000. Introduction to AI robotics. MIT press.

[2] Niku, S. 2010. Introduction to robotics. John Wiley & Sons.

[3] Nehmzow, Ulrich. 2012. Mobile robotics : A practical introduction. London: Praxis.

[4] Alves S. F., Rosario J. M., Ferasoli Filho H., Rincon L. K., & Yamasaki R. A. 2011. Conceptual bases of robot navigation modeling, control and applications. In Advances in Robot Navigation. InTech.

[5] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. 2014;2011;. Introduction to autonomous mobile robots. 2nd ed. Cambridge: MIT Press.

[6] Goris, K. 2005. Autonomous mobile robot mechanical design. VrijeUniversiteitBrussel, Engineering Degree Thesis, Brussels, Belgium.

[7] Özkil A. G. 2009. Technical Report on Autonomous Mobile Robot Navigation.

[8] I brahim, M. Y., and A. Fernandes. 2004. Study on mobile robot navigation techniques.

[9] Nirmala, G., Dr S. Geetha, and Dr S. Selvakumar. 2017. Mobile robot localization and navigation in artificial intelligence: Survey. Computational Methods in Social Sciences 4 (2): 12-22.

[10] Beni, Gerardo. 2005. From swarm intelligence to swarm robotics. In . Vol. 3342, 1-9. Berlin, Heidelberg: Springer Berlin Heidelberg.

[11] Şahin, Erol. 2005. Swarm robotics: From sources of inspiration to domains of application. In . Vol. 3342, 10-20. Berlin, Heidelberg: Springer Berlin Heidelberg.

[12] Brutschy, A. 2009. Task allocation in swarm robotics. Towards a method for selforganized allocation to complex tasks. University Libre de Brux-elles, 1050 Bruxelles, Belgium, Technical Report TRlIRIDIA12009–007, 52009.

[13] Rashid, Razif, Irraivan Elamvazuthi, Mumtaj Begam, and M. Arrofiq. 2010. Differential drive wheeled mobile robot (WMR) control using fuzzy logic techniques.

[14] Rekik, Chokri, Mohamed Jallouli, and Nabil Derbel. 2009. Integrated genetic algorithms and fuzzy control approach for optimization mobile robot navigation.

[15] Faisal, Mohammed, Ramdane Hedjar, Mansour Al Sulaiman, and Khalid Al-Mutib. 2013. Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. International Journal of Advanced Robotic Systems 10 (1): 37.

[16] Narvydas, G., R. Simutis, and V. Raudonis. 2007. Autonomous mobile robot control using fuzzy logic and genetic algorithm.

[17] Güzel, Mehmet Serdar, Mehmet Kara, and Mehmet Sıtkı Beyazkılıç. 2017. An adaptive framework for mobile robot navigation. Adaptive Behavior 25 (1): 30-9.

[18] Hayes, A. T., and P. Dormiani-Tabatabaei. 2002. Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots.

[19] Nga Le Thi Thuy, and Thang Nguyen Trong. 2017. The multitasking system of swarm robot based on null-space-behavioral control combined with fuzzy logic. Micromachines 8 (12): 357.

[20] Ducatelle, Frederick, Gianni A. Di Caro, Carlo Pinciroli, Francesco Mondada, and Luca Gambardella. 2011. Communication assisted navigation in robotic swarms: Self-organization and cooperation.

[21] Sugawara, K., T. Kazama, and T. Watanabe. 2004. Foraging behavior of interacting robots with virtual pheromone.

[22] Batalin, Maxim A., and Gaurav S. Sukhatme. 2004. Coverage, exploration and deployment by a mobile robot and communication network. Telecommunication Systems 26 (2): 181-96.

[23] Houcque, David. "Introduction to Matlab for engineering students." Northwestern University (2005): 1-64.

[24] “Math Works”, [Online]. Available: https://www.mathworks.com. Accessed: September 2018.

[25] “Coppelia Robotics”, [Online]. Available: http://www.coppeliarobotics.com. Accessed: September 2018.

[26] Mondada, F., Franzi, E., & Guignard, A.1999. The development of khepera. In Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop No. LSRO-CONF-2006-060: 7-14.

[27] “K Team ”, [Online]. Available: https://www.k-team.com. Accessed: March 2018.

AUTHOR

Sherif Kamel Hussein Hassan Ratib: Graduated from the faculty of engineering in 1989 Communications and Electronics Department,Helwan University. He received his Diploma,MSc,and Doctorate in Computer Science-2007, Major Information Technology and Networking. He has been working in many private and governmental universities inside and outside Egypt for almost 15 years.He shared in the development of many industrial courses. His research interest is GSM Based Control and Macro mobility based Mobile IP. He is an Associate Professor and Faculty Member at Communications and Computer Engineering department in October University for Modern Sciences and Arts - Egypt. Now he is working as head of Computer Science department in Arab East Colleges for Postgraduate Studies in Riyadh- KSA.