Multi-Agent System with Fuzzy Logic Control for Autonomous Mobile Robots in Known Environments

Cinthya Solano-Aragón[1] and Arnulfo Alanis[2]

[1] Student of Master of Science in Computer Science, Instituto Tecnologico de Tijuana de Tijuana, Calzada Tecnologico S / N, Tijuana, Mexico
cinthyasolanoa@gmail.com
[2] Division of Graduate Studies and Research, Instituto Tecnologico de Tijuana, Calzada Tecnologico, S/N, Tijuana, Mexico
alanis@tectijuana.edu.mx

Abstract. This paper describes the development of a Multi-Agent System (MAS), which is supported with fuzzy logic (to control the robots movements in a reactive path) and vision, which controls an autonomous mobile robot to exit a maze. The research consists of two stages. In the first stage the problem is to be able to make the robot exit a maze, the mobile robot is positioned at the entrance (point A) and should reach an output (B). It should be noted that we are working with a NXT Lego MINDSTORMS robot. In its second phase the problem is to make the robot search for a recognized object, for this, a camera is used to capture images, which will be processed with vision techniques, for their identification, and after that, the SMA takes the decision to evade or take the object as appropriate.

1 Introduction

Making smarter robots is a problem that has captivated the scientific community for several years now, this being a big challenge to overcome even for man. To move from one place to another from an initial to a final point with only the information of what we want to reach at the end is a complicated task. The problem has been solved with soft computing methods, such as fuzzy logic, neural networks, genetic algorithms, hybrid systems, among others. Navigation of the robot has been achieved taking a completely controlled environment where we have total knowledge of the world [8, 11, 15, 14].

Human beings have the ability to react to unexpected situations and the ability to use their reasoning to react to situations that arise, an example is a situation of traffic management, to which we think that we have the best route and with this the rest of the trip is done reacting to what was observed. This could be, at a traffic light, when it changes from green to yellow, the reaction will be learning, acceleration or deceleration, a situation that when you start, it has to be learned.

O. Castillo et al. (Eds.): Evolutionary Design of Intelligent Systems, SCI 257, pp. 33–52.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

Knowing that we have this ability to react, we will use fuzzy logic to transfer the knowledge we use to carry out this process by establishing computer-type fuzzy "if-then" rules and exploiting the advantages offered to us by fuzzy logic, which is the use of linguistic variables [26].

A man driving a car, as in many other activities, derives its information by looking at distances, clearances and other identifying information that we collect through the light. In this research we use vision techniques to draw from that vital information for navigation control, supported with special sensors. An example is the ultrasonic sensors that are used to measure distance and the light sensors that we can use to measure the change in light intensity, all of these help to extract environmental information necessary for decision making.

2 Agents

Let's first deal with the notion of intelligent agents. These are generally defined as "software entities", which assist their users and act on their behalf. Agents make your life easier, save you time, and simplify the growing complexity of the world, acting like a personal secretary, assistant, or personal advisor, who learns what you like and can anticipate what you want or need. The principle of such intelligence is practically the same of human intelligence. Through a relation of collaboration-interaction with its user, the agent is able to learn from himself, from the external world and even from other agents, and consequently act autonomously from the user, adapt itself to the multiplicity of experiences and change its behavior according to them. The possibilities offered for humans, in a world whose complexity is growing exponentially, are enormous [20][19][7][18].

We need to be careful to distinguish between rationality and omniscience. An omniscient agent knows the actual outcome of its actions, and can act accordingly; but omniscience is impossible in reality. Consider the following example: I am walking along the Champs Elys´ees one day and I see an old friend across the street. There is no traffic nearby and I'm not otherwise engaged, so, being rational, I start to cross the street. Meanwhile, at 33,000 feet, a cargo door falls off a passing airliner, and before I make it to the other side of the street I am flattened. Was I irrational to cross the street? It is unlikely that my obituary would read "Idiot attempts to cross street."Rather, this point out that rationality is concerned with expected success given what has been perceived. Crossing the street was rational because most of the time the crossing would be successful, and there was no way I could have foreseen the falling door. Note that another agent that was equipped with radar for detecting falling doors or a steel cage strong enough to repel them would be more successful, but it would not be any more rational [23].

3 FIPA

FIPA (The Foundation of Intelligence Physical Agents) specifications represent a collection of standards, which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. The life cycle [46] of specifications details what stages a specification can attain while it is part of the FIPA standards process. Each specification is assigned a specification identifier [47] as it enters the FIPA specification life cycle. The specifications themselves can be found in a Repository [11]. The Foundation of Intelligent Physical Agents (FIPA) is now an official IEEE Standards Committee. The UML modeling language is the most popular system used today and is a graphic language to visualize, specify, build and document a system. Gaia is a methodology for application development in the paradigm of agents, one of the most curious aspects of Gaia, is the fact that the requirements specification is completely independent of the analysis and design.

4 Proposed Method

To realize this work, the creation of a SMA is needed, which is composed of three agents, an agent for the management of sensors, an agent for handling the drive and a last agent to act as coordinator of the multi-agent system.

Summarizing the SMA is composed of the following agents:

Node Agent (NA), a set of sensors.
Task Agent (TA), a set of servo motors.
Agent System (AS), coordinator of the system [1].

The Node Agent (NA) will be responsible for the sensors used by the robot, which are:

- Ultrasonic Sensor
- Light Sensor left
- Light Sensor right
- Camera

These sensors are used to interact with the world to achieve our goal.

The Task Agent (TA) will be used to drive to move forward, rewind, or make some movement to escape.

The System Agent (SA) will be responsible for coordinating the other actors, the NA and TA, and will also have activities not only the coordination, which will be to recognize the object that is opposite, calculate the angle of the object, which will help us to know that will guide the robot. Depending on the recognition that was obtained, whether or not the object, which will tell us that there is no escape or the object based on this decision was to trigger the switch which we will control the outcome to be used, reactive or path to better performance of the controllers.

In Fig. 1 we show graphically the architecture of the complete system.

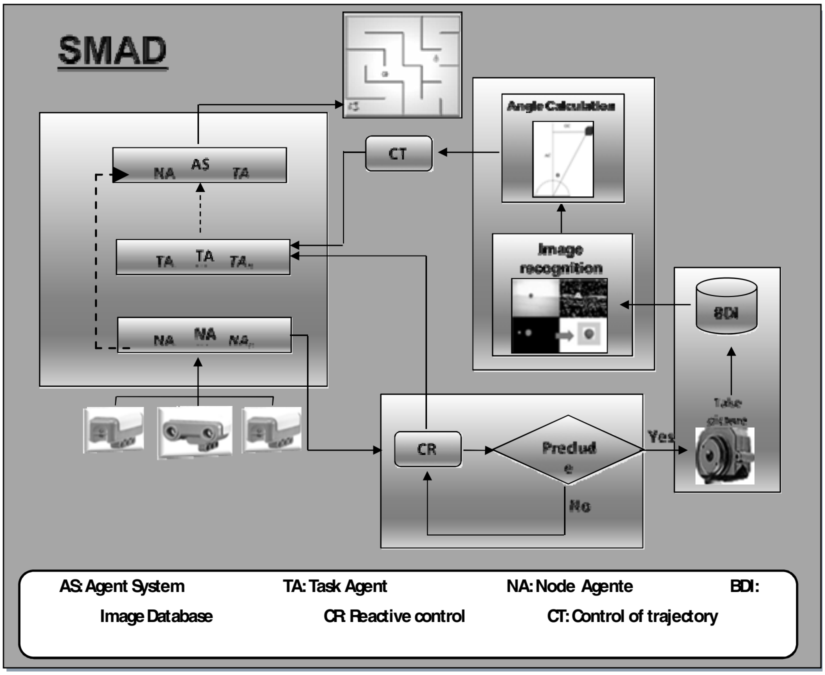


Fig. 1. Architecture of the complete system.

We can describe the proposed method as follows:

In the operation of the model, we have 2 main blocks, which are responsible for knowledge and learning (with the paradigm of intelligent agents) and the vision and control (using fuzzy logic).

These modules are described below, the first module contains 3 agents, NA [Node Agent], TA [Task Agent] SA [System Agent] Agent System (AS), and will know every time the operation of the other agents.

To detect a change in the environment the Agent Node [that is in charge of the sensors (ultrasonic, camera and two light sensors)] with the ultrasonic sensor and two light sensors, starts its operation, and we start at the state called reactive control; this because you have to move and sense all the time until it finds an obstacle, this can happen once a photo is taken, which will be sent to the database of images (BDI), the image will be applied a pre-processing for recognition in order to know whether the object is found, this decision was taken on the angle (angle to take the decision to bypass the object), able to escape and move forward, trend data, they are caught in the trajectory control process, to observe the speed values, the Task Agent (which is responsible for the drive), once all the parameters necessary for the performance of the robot agent system (AS) who is the coordinator, and executes instructions in the robot.

For the development of the two key tools we used Gaia and UML for the analysis and design of agents. The completion of the Multi-Agent System (MAS) is based on the FIPA standards for better utilization and greater possibility of extension.

5 Analysis and Design of the Multi-Agent System (MAS)

The first activity was the analysis and design of the multi-agent system as mentioned above and to develop the two tools and we describe as follows.

5.1 UML

Use Case Diagrams

The Use Case diagrams show the granularity of the system into reusable pieces of functionality, interaction of players with the functionality of the system, visually organize user requirements and allow the contract to certify the functionality, formalize the process map.

Fig. 2 presents the use case of the sense in which the distance is appreciated that all other processes must be carried out before and after to complete. In Fig. 3 we show the use case of the motor A move, which is seen in all other processes that must be carried out before and after to complete [2][3][4][5][6].

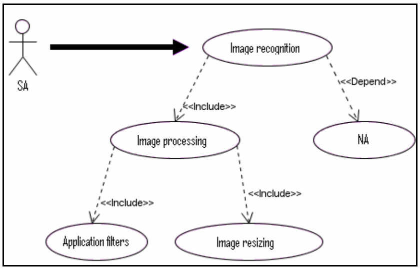


Fig. 2. Use case diagram of the process of image recognition.

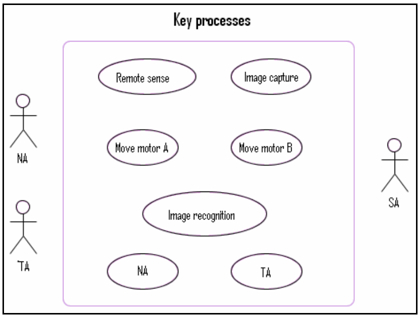


Fig. 3. Main processes of use cases.

Fig. 4 presents the case for use of the remote sensing process, which shows that other processes must be carried out before and after to complete. Fig. 5 shows the use case of the motor A move, which is seen in all other processes must be carried out before and after to complete.

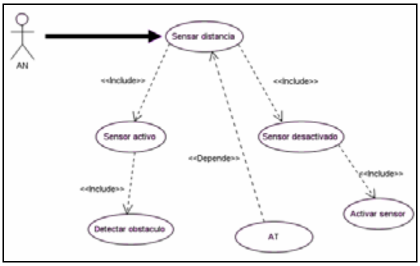


Fig. 4. Use case diagram of the remote sense.

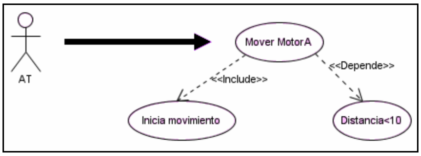


Fig. 5. Use case diagram of the process moving motor A.

Sequence diagrams describe the interaction of objects that require the functionality of the different scenarios of a use case, objects are represented with their life cycle within a time series, and each possible scenario of a use case can be represented as a sequence diagram.

Below in Fig. 6 we show the sequence diagram of the use case moving motor A. In Fig. 7 presents the sequence diagram of use case to capture the image which is carried out by the agent node (AN).

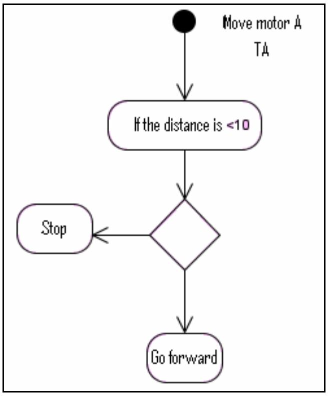


Fig. 6. Sequence diagram use case move motor A.

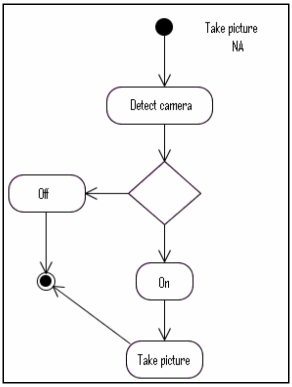


Fig. 7. Sequence diagram use case capture image.

In Fig. 8 presents the sequence diagram used for the case sense of distance, which is carried out by the agent node (AN).

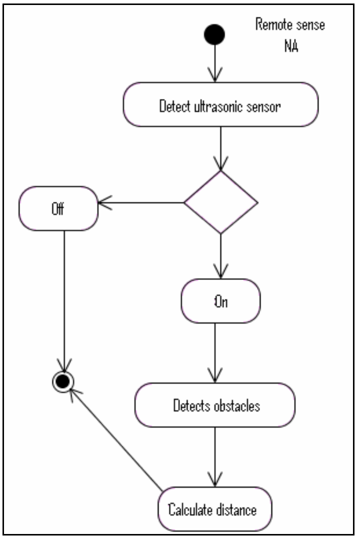


Fig. 8. Sequence diagram use case sense distance.

5.2 Gaia

Below we show the diagrams in which we can observe the responsibilities, permissions, activities and protocols to be followed by each agent in the Multi-Agent System (MAS).

Agent node

Responsibilities:

Sense:

Active

Inactive

Permissions:

Ultrasonic sensor: measures distances from 0 cm to 255 cm, with a delay of one millisecond signal.

Camera Sensor: take pictures in an estimated time of seconds, saves all images in a folder for processing. It has a degree of vision approximately of 50 degrees.

Light sensors: they measure the intensity with which an object reflects light. This makes a light emitting and measuring the portion of the return is received, the table 3.2 shows the ranges of values.

Activities:

The only activities are sensing and taking pictures of the world.

Protocols:

Ultrasound: This is responsible for sensing the distance to get to an obstacle. The initiative for this would be the agent system, which indicates the time of initiation.

Chamber is responsible for obtaining the images within the scene, and as for the ultrasonic sensor system depends for its initiation.

Light sensors: the role of these sensors is to measure the intensity with which an object reflects light (walls, diagrams). This makes a light emitting and measuring the portion of the return is received and handled the ranges are 0 to 1023 RAW.

Agent system

Responsibilities:

It is responsible for image processing and decision making, and sends a message to agent communication process completed?

Permissions:

This is the one that has full access, sensors, communication, engine and handling agents and NXT in general.

Activities:

Making decisions based on the behavior and implements what is needed to finish the job.

Protocols:

Full Access, sensors, communication, engine and handling agents and NXT in general, the entry for this information would be provided by the agents, getting a response from these so they generate an output and generate interaction among them.

Task agent

Responsibilities:

Motor movement

Permission:

Depends entirely on the agent system to perform its task.

Activities:

Sensors for the engine: they measure in degrees is given for each engine is 360 degrees for one revolution, a revolution for the tire of each motor, the motors can rotate independently, can also be used with constant speed motors.

Protocols:

The system is the initiator, resulting in a movement or departure, which can be any direction, depending on the decision taken by the agent system.

Figure 9 shows the three models that form the multi-agent system (MAS) using Gaia.

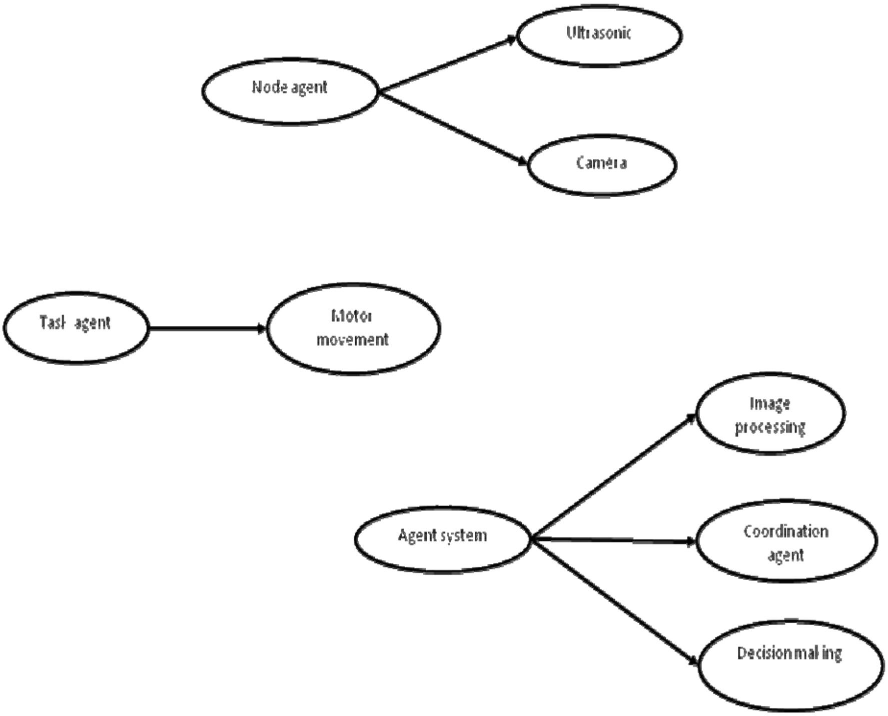


Fig. 9. Shows the diagrams of the Agent Node (AN), Task Agent (TA), and Agent System (AS) in Gaia.

Fig. 10. Diagram of the relationship model of the SMA.

Fig. 10 presents the relationship between the agents of the multi-agent system (MAS).

5.3 Knowledge Base

Below we show the knowledge base that the MAS uses to function properly.

Facts

NA$_i$.SU. STATE =(X), where X can have 2 states (ON, OFF)
NA$_i$.SU.OBJ=(Y) where Y can have 2 states (DETECTS, NO DETECTS)
NA$_i$.SC. STATE = (Z) where Z can have 2 states (ON, OFF)
NA$_i$.SC.TIMAGE
NA$_i$.SLI. STATE = (W) where W can have 2 states (ON, OFF)
NA$_i$.SLD. STATE = (V) where V can have 2 states (ON, OFF)
TA$_j$.M.GO(evaluationFIS[SU, SLI, SLD]) where SU,SLI,SLD FIS are entries for return rates of both servomotors (MA, MB)
TA$_j$.M.STOP
TA$_j$.MA.GO(evaluationFIS[SU, SLI, SLD]) where SU,SLI,SLD FIS are entries for return rates of both servomotors (MA, MB)
TA$_j$.MA.DECREASEV
TA$_j$.MB.GO(evaluationFIS[SU, SLI, SLD]) where SU,SLI,SLD FIS are entries for return rates of both servomotors (MA, MB)
TA$_j$.MB. DECREASEV
SA. BEGINTASK
SA.RECONOCIMAGE.STATE = (N) where N can have 2 states (YES, NO)
SA.ACTION.TAKEFIGURE
SA.ACTION.AVOID
SA.FIGUREFOUND
SA.ENDMAZE = (M) where M can have 2 states (YES, NO)
SA.ENDTASK

Rules

If SA. BEGINTASK and NA$_i$.*SU.STATE=OFF* then NA$_i$.*SU.STATE=ON*
If SA. BEGINTASK and NA$_i$.*SC.STATE=OFF* then NA$_i$.*SC.STATE=ON*
If SA. BEGINTASK and NA$_i$.*SLI.STATE=OFF* then NA$_i$.*SLI.STATE=ON*
If SA. BEGINTASK and NA$_i$.*SLD.STATE=OFF* then NAi.*SLD.STATE=ON*
If SA.ENDTASK and NA$_i$.*SU.STATE=ON* then NA$_i$.*SU.STATE=OFF*
If SA.ENDTASK and NA$_i$.*SC.STATE=ON* then NA$_i$.*SC.STATE=OFF*
If SA.ENDTASK and NA$_i$.*SLI.STATE=ON* then NA$_i$.*SLI.STATE=OFF*

If SA.ENDTASK and NA$_i$.*SLD.STATE=ON* then NA$_i$.*SLD.STATE=OFF*

If SA. BEGINTASK and NA$_i$.*SU.STATE=ON* then TA$_j$.M.GO(evaluationFIS[SU, SLI, SLD])

If TA$_j$.M.GO(evaluationFIS[SU, SLI, SLD]) and NA$_i$.*SU.OBJ=* DETECTS then NA$_i$.*SC.TIMAGE*

If NA$_i$.*SC.TIMAGE* then SA.RECONOCIMAGE

If SA.RECONOCIMAGE.STATE = SI then SA.ACTION.TOMAFIG

If SA.RECONOCIMAGE.STATE = NO then SA.ACTION.EVADIR

If SA.ACTION.TAKEFIGURE then TA$_j$.MA.GO(evaluationFIS[SU, SLI, SLD]) and TA$_j$.MB.GO(evaluationFIS[SU, SLI, SLD])

If SA.ACTION.AVOID then TA$_j$.MA.GO(evaluationFIS[SU, SLI, SLD]) and TA$_j$.MB.GO(evaluationFIS[SU, SLI, SLD])

If SA.ACTION.TAKEFIGURE and SA.ENDMAZE = YES then SA.ENDTASK

If SA.ACTION.TAKEFIGURE and SA.ENDMAZE = NO then TA$_j$.MA.GO(evaluationFIS[SU, SLI, SLD]) and TA$_j$.MB.GO(evaluationFIS[SU, SLI, SLD])

If SA.ACTION.AVOID and SA.ENDMAZE = YES then TA$_j$.MA.GO(evaluationFIS[SU, SLI, SLD]) and TA$_j$.MB.GO(evaluationFIS[SU, SLI, SLD])

6 Simulation

Figure 11 shows the plant and the controller for the simulations that were carried out in Simulink of Matlab 2007b.

The controller is based on the following fuzzy system, which contains the rules that work with the robot.

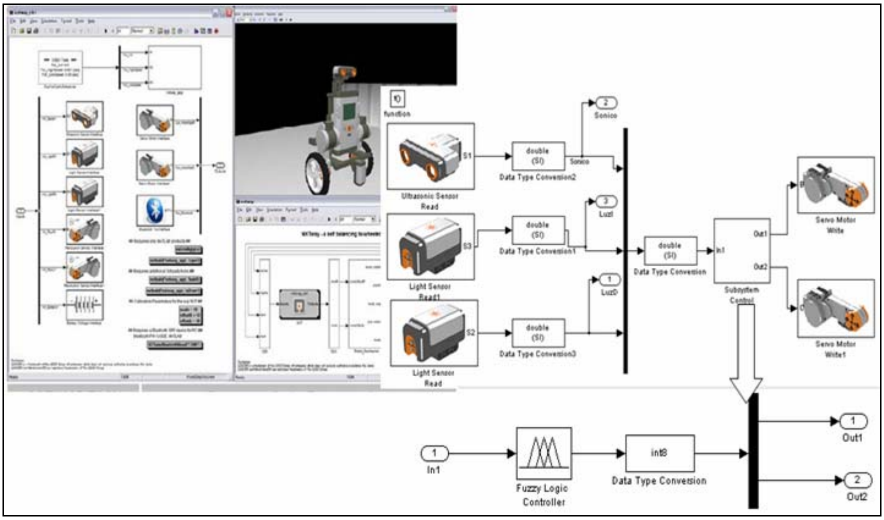


Fig. 11. Plant for performing simulations.

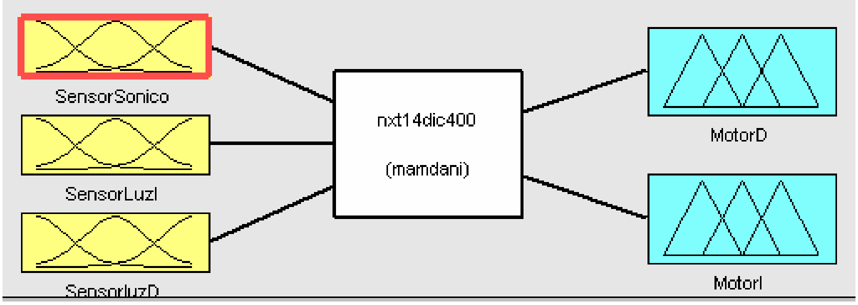


Fig. 12. Structure of the fuzzy system that works for the simulations.

The fuzzy system has three inputs, two outputs and consists of ten rules for inference.

The first input variable of the fuzzy system is the ultrasonic sensor, which has three membership functions, which are linguistic (close, near, far), as shown in Fig. 13.

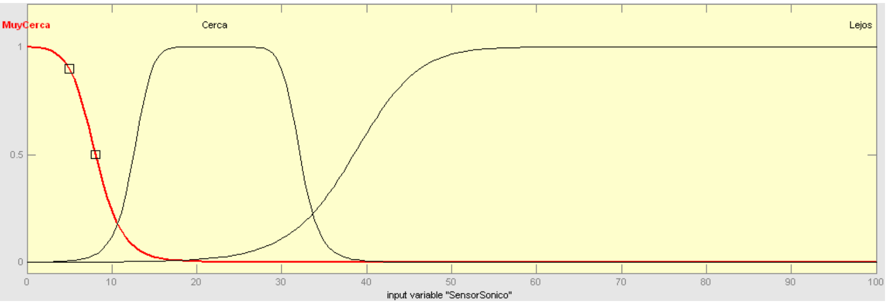


Fig. 13. Membership functions for the ultrasonic sensor.

The second variable of the fuzzy system is the light sensor that has two membership functions that are free and wall, as shown in Fig. 14.

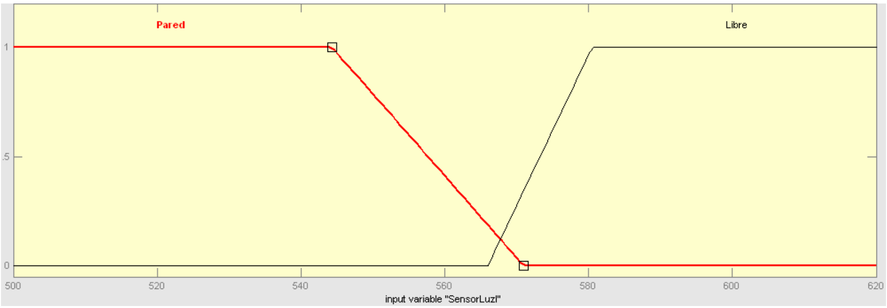


Fig. 14. Membership functions of the left light sensor.

The third variable of the fuzzy system is the light sensor that has two membership functions that are free and wall, as shown in Fig. 15.

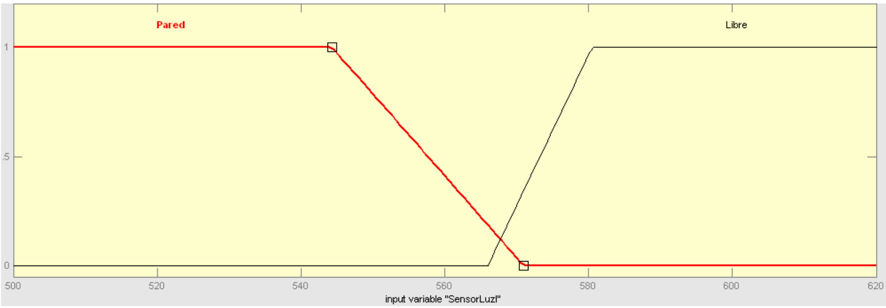


Fig. 15. Membership functions of the right light sensor.

This is the first output variable is the speed of the left engine, has five membership functions as shown in Fig. 16.

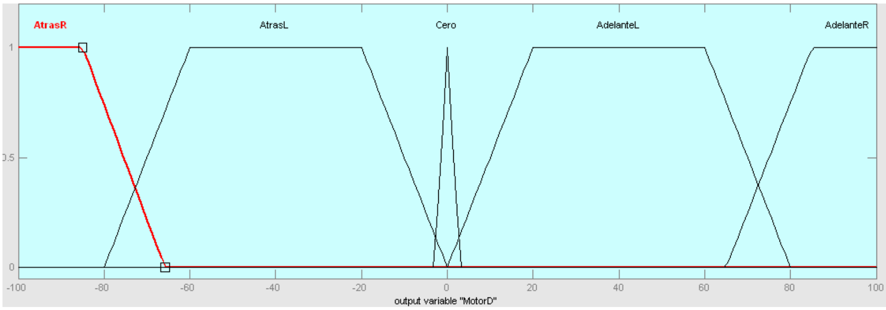


Fig. 16. Membership functions of the left motor.

This is the second output variable is the speed of the right engine, has five membership functions as shown in Fig. 17.

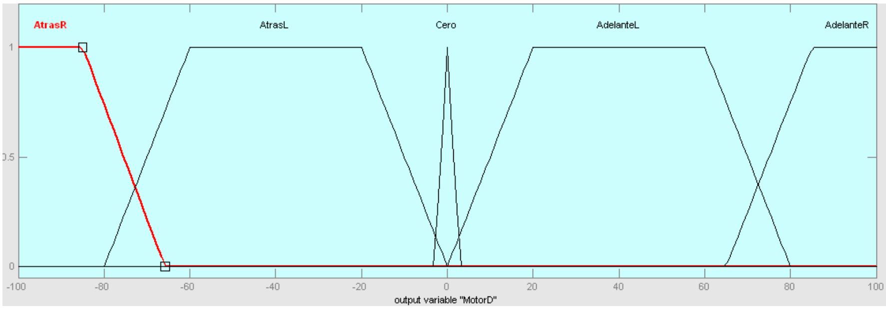
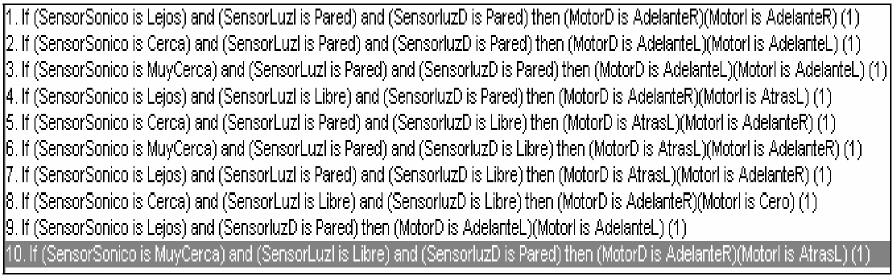


Fig. 17. Membership functions of the right motor.

In Fig. 18 we show the rules that are used in the system, which are 10 fuzzy rules.



1. If (SensorSonico is Lejos) and (SensorLuzI is Pared) and (SensorluzD is Pared) then (MotorD is AdelanteR)(MotorI is AdelanteR) (1)
2. If (SensorSonico is Cerca) and (SensorLuzI is Pared) and (SensorluzD is Pared) then (MotorD is AdelanteL)(MotorI is AdelanteL) (1)
3. If (SensorSonico is MuyCerca) and (SensorLuzI is Pared) and (SensorluzD is Pared) then (MotorD is AdelanteL)(MotorI is AdelanteL) (1)
4. If (SensorSonico is Lejos) and (SensorLuzI is Libre) and (SensorluzD is Pared) then (MotorD is AdelanteR)(MotorI is AtrasL) (1)
5. If (SensorSonico is Cerca) and (SensorLuzI is Pared) and (SensorluzD is Libre) then (MotorD is AtrasL)(MotorI is AdelanteR) (1)
6. If (SensorSonico is MuyCerca) and (SensorLuzI is Pared) and (SensorluzD is Libre) then (MotorD is AtrasL)(MotorI is AdelanteR) (1)
7. If (SensorSonico is Lejos) and (SensorLuzI is Pared) and (SensorluzD is Libre) then (MotorD is AtrasL)(MotorI is AdelanteR) (1)
8. If (SensorSonico is Cerca) and (SensorLuzI is Libre) and (SensorluzD is Libre) then (MotorD is AdelanteR)(MotorI is Cero) (1)
9. If (SensorSonico is Lejos) and (SensorluzD is Pared) then (MotorD is AdelanteL)(MotorI is AdelanteL) (1)
10. If (SensorSonico is MuyCerca) and (SensorLuzI is Libre) and (SensorluzD is Pared) then (MotorD is AdelanteR)(MotorI is AtrasL) (1)

Fig. 18. Rules of the fuzzy system (FIS).

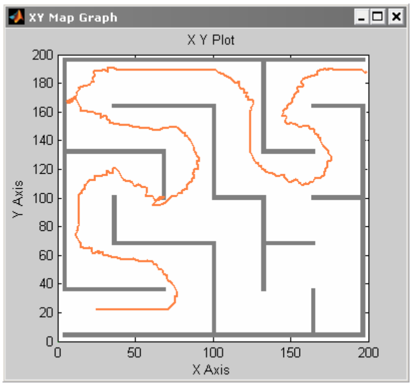


Fig. 19. Simulation1 with a duration of 00:63 minutes.

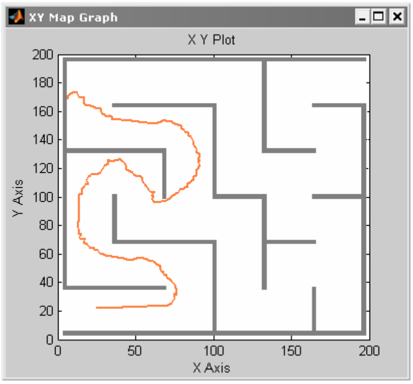


Fig. 20. Simulation2 with a duration of 01:30 minutes.

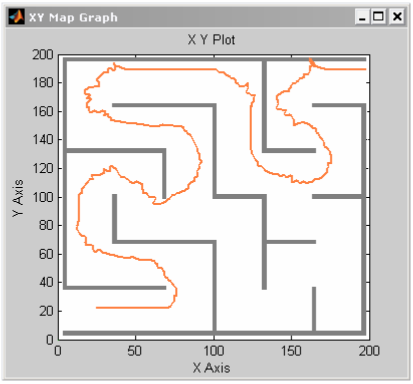


Fig. 21. Simulation3 with a duration of 00:51 minutes.

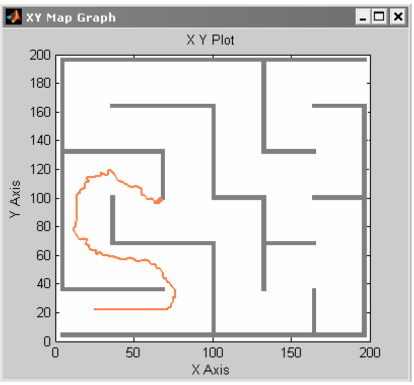


Fig. 22. Simulation4 with a duration of 00:50 minutes.

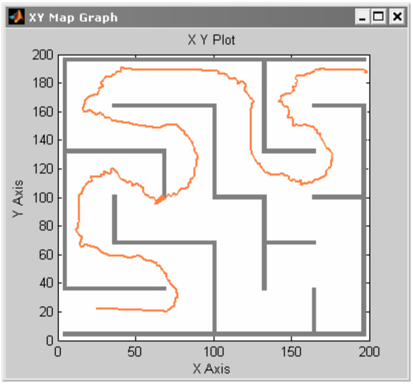


Fig. 23. Simulation5 with a duration 00:50 minutes.

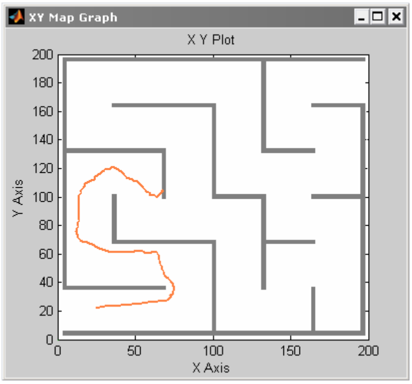


Fig. 24. Simulation6 with a duration of 00:17 minutes.

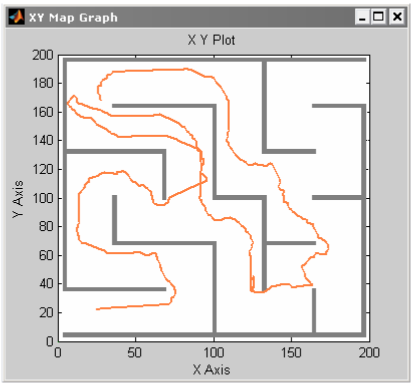


Fig. 25. Simulation7 with a duration of 00:57 minutes.

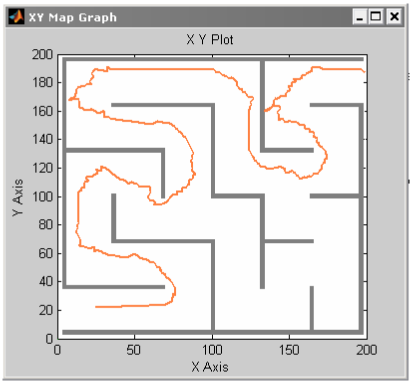


Fig. 26. Simulation8 with a duration of 00:52 minutes.

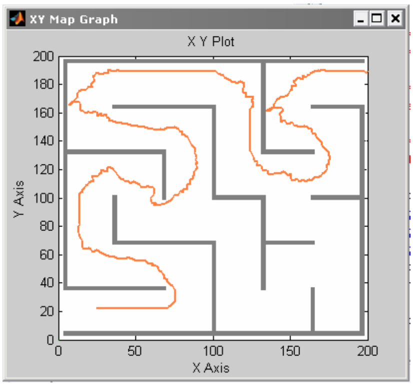


Fig. 27. Simulation9 with a duration of 00:52 minutes.

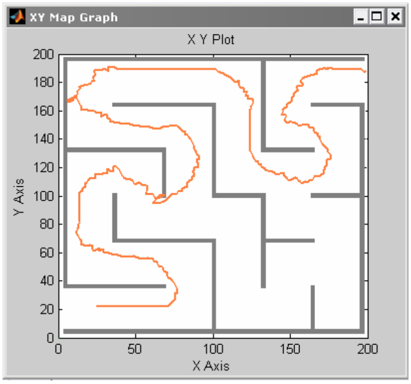


Fig. 28. Simulation10 with a duration of 00:51 minutes.

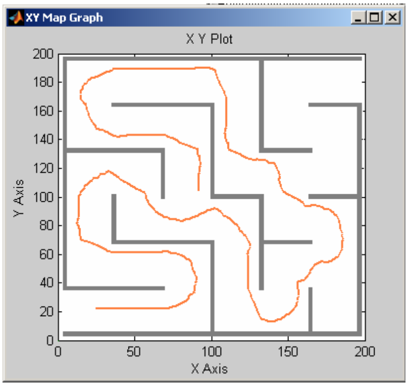


Fig. 29. Simulation11 with a duration of 01:25 minutes.

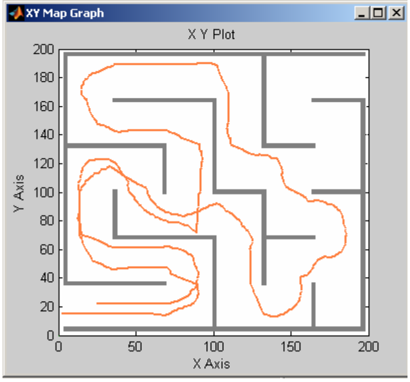


Fig. 30. Simulation12 with a duration 01:46 minutes.

Table 1 contains the information from the simulations regarding the duration that each one had, and whether or not out the robot was able to get out of the labyrinth, it also highlights the best simulation time.

Table 1. A comparison between the simulations.

Number of Simulation	Duration/minutes	Exit the maze
1	00:63	Yes
2	1:30	No
3	00:51	Yes
4	00:50	No
5	00:50	Yes
6	00:17	No
7	00:57	No
8	00:52	Yes
9	00:52	Yes
10	00:51	Yes
11	1:25	No
12	1:46	No

7 Conclusions

There is a big diversity of intelligent applications, in this particular case, the implementation of multi-agent systems is complex because it is short, and it incorporates other techniques, making it even more complex.

One of the main objectives of this research initially was to open a new view of intelligent agents using fuzzy logic, which hitherto has not been given the approach that is intended here. Fuzzy logic gives a degree of utilization of this paradigm of intelligent agents, but more research is needed in order to defend this idea as innovative.

References

[1] Alanis, A.: Contribution to the design of Fault Tolerant Distributed Systems for industrial control: proposal for a new paradigm based on Intelligent Agents, PhD, Universidad Politecnica de Valencia (November 2007)

[2] Alanis, A., López, M., Muñoz, R., Pulido, M., Martínez, P., Beltrán, M.: Multi-agent System for a Lego NXT robot to access the collaboration object 1.8 º. In: National Congress of Electrical and Electronics Engineering of the Mayab conieem 2008 Instituto Tecnológico de Mérida, to be held in the city of Merida, Yucatan, Mexico, April 21-25 (2008)

[3] Alanis, A., López, M., Muñoz, R., Pulido, M., Martínez, P., Beltrán, M.: Multi-agent system for a Lego NXT robot to access the collaboration of objects 1. In: XXIX International Congress of Engineering in electronics, electro 2008, Instituto Tecnológico de Chihuahua, division of graduate studies and research, Chihuahua, Chih. Mexico, October 29-31 (2008)

[4] Alanis, A., López, M., Parra, B., Serrano, M., Ayala, E., Solano, C.: Multi-agent system for search and object recognition using vision, in a Lego NXT robot 1.8 º. In: National Congress of Electrical and Electronics Engineering of the Mayab conieem 2008 Instituto Tecnológico de Mérida, to be held in the city of Merida, Yucatan, Mexico, April 21-25 (2008)

[5] Alanis, A., López, M., Parra, B., Serrano, M., Ayala, E., Solano, C.: Multi-agent system for search and object recognition using vision, in a Lego NXT robot 1. In: XXIX International Congress of Engineering in electronics, electro 2008 Instituto Tecnologico de Chihuahua, division of graduate studies and research, Chihuahua, Chih. Mexico, October 29, 31 (2008)

[6] Antoquia University, Faculty of Engineering, "Neural Networks" (June 2003)

[7] Bratko, I.: Prolog for Programming Artificial Intelligence. Addison-Wesley, Reading (1986)

[8] Cupertino, F., Giordano, V., Naso, D., Delfine, L.: Fuzzy control of a mobile robot. IEEE Robotics & Automation Magazine, 74–81

[9] Galván, R., Monsivais, A.: Resolution of mazes with Lego Mindstorms NXT with LabVIEW 7.1, thesis work ITT

[10] Dhananjay, G.S.: ANN Models for Speaker Recognition Based on Difference Cepstrals. Indian Institute of Technology, Madras (2000)

[11] Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing for computational approach to learning and machine intelligence. Prentice Hall, Englewood Cliffs (1997)

[12] Jang, Sun, J., Mizutani, J.: Neuro-Fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Englewood Cliffs (1997)

[13] López, A.: Program Development in Matlab, Matlab. no. Bulletin of the Cuban Society of Mathematics and Computer Science 2(2), 105–112 (2004)

[14] Maček, K., Petrović, I., Siegwart, R.: A control method for stable and smooth path following of mobile robots. In: Proceedings of the 2nd European Conference on Mobile Robots - ECMR 2005, Ancona, Italy, September 7-10, pp. 128–133 (2005)

[15] Leyden, M., Toal, D., Flanagan, C.: A Fuzzy Logic Based Navigation System for a Mobile Robot. In: Proceedings of Automatisierungs Symposium, Wismar, Germany (1999)

[16] Melendez, A.: Control and monitoring reagent for a mobile robot using fuzzy logic, thesis work, ITT (August 2008)

[17] Morgan, D.P., Scofield, C.L.: Neural Networks and Speech Processing. Kluwer Academic Publishers, Dordrecht (1991)

[18] Norving, P., Russell, S.: Artificial intelligence a modern approach. Prentice Hall, Australia (1996)

[19] Etzioni, O., Lesh, N., Segal, R.: Bulding for Softbots UNIX (preliminary report). Tech. Report 93-09-01. Univ. of Washington, Seattle (1993)

[20] Cohen, P.R., et al.: An Open Agent Architecture. In: Working Notes of the AAAI Spring symp.: Software Agent, pp. 1–8. AAAI Press, Cambridge (1994)

[21] Russell, S., Norvig, P.: Intelligent Agent. In: Artificial Intelligence to Modern Aproach. Pretence artificial Hall series in intelligence, pp. 31–52 (1994)

[22] Sánchez, O.: Voice and image recognition by neural networks to guide a robot, thesis work ITT (October 2008)

[23] Suárez, J.: Location and tracking of trajectories with robots Wayfarer in natural surroundings, thesis work Universidad Complutense de Madrid

[24] University of Vigo, Department of Computer Science, Problems of search and optimization (October 2004)

[25] Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics 31(6), 891–901 (2001)

[26] Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, and 3, Information Sciences (1975)

[27] Intelligent Autonomous Robots are the new generation, September 3 (2008), `http://www.tendencias21.net/Los-Robots-Inteligentes-Autonomos-son-la-nueva-generacion_a744.html`

[28] Introduction to intelligent control, September 3 (2008), `http://isa.umh.es/isa/es/asignaturas/cas/TRANSPCI.PDF`

[29] What is Simulink?, September 4 (2008), `http://voltio.ujaen.es/jaguilar/matlab/Manual20Matlab_Simulink/manual%%20simulink/SIM_01%20-%20Que_%20es.htm`

[30] Simulation of a fuzzy system for controlling engine speed of a CD, September 8 (2008), `http://www.uaem.mx/cicos/memorias/3ercic2004/Articulos/articulo3.pdf`

[31] fuzzy systems, September 8 (2008), `http://ants.dif.um.es/staff/juanbot/ml/files/20022003/fuzzy.pdf`

[32] Control of local mobile robots based on statistical methods and genetic algorithms, October 1 (2008), `http://www.dccia.ua.es/ria/artics/caepia97.pdf`

[33] The NXT Bluetooth C + + library, October 6 (2008), `http://www.norgesgade14.dk/bluetoothlibrary.php`

[34] Embedded Coder Robot NXT, October 6 (2008), `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=13399`

[35] Tools for programming free agents, October 13 (2008), `http://torio.unileon.es/~vmo/pubs/waf00.pdf`

[36] Analysis and design of reinforced learning agents, October 13 (2008), http://www.itcm.edu.mx/mccc06/publicaciones/Analisisydise%F1odeagentes.pdf
[37] An architecture for mobile robots small, October 13 (2008), http://igarrido.vtrbandaancha.net/papers/gbot_arquitectura.pdf
[38] Class diagrams, UML artifacts, October 22 (2008), http://www.vico.org/aRecursosPrivats/UML_TRAD/talleres/mapas/UMLTRAD_101A/LinkedDocuments/UML_diagClases.pdf
[39] Altamirano, R.L.: Laboratory of Computer Vision, http://ccc.inaoep.mx/labvision~/vcomp.htm
[40] Maldonado, O.: Republic of Knowledge, Computer Vision. p. 3, http://www.depi.itch.edu.mx/apacheco/expo/html/ai11/vision.html
[41] Salinas, R.: Neural Network Architecture Parametric Face Recognition, University of Santiago de Chile, http://cabierta.uchile.cl/revista/17/articulos/pdf/paper4.pdf
[42] http://www.fipa.org/specifications/lifecycle.html
[43] http://www.fipa.org/specifications/identifiers.html