Humanoid robot using Petri Nets as tool for decision making

Federico Furlán, Elsa Rubio, Humberto Sossa and Víctor Ponce

Centro de Investigación en Computación
Instituto Politécnico Nacional, Av. Juan de Dios Bátiz
S/N, Gustavo A. Madero, c.p. 07738. Ciudad de México, México
(e-mail: ffurlan_b11@sagitario.cic.ipn.mx,
{erubio, hsossa, vponce}@cic.ipn.mx).

Abstract: This paper presents a supervisory control system for humanoid robot motion planning. The proposed system is a supervisory structure formed by two hierarchical levels of a discrete event system. The high level system is represented by a Petri net. This Petri net behaves as a supervisor that indicates the sequence of motions that the robot has to make. A robot walking in a closed space forms the lower level. The robot decisions are modeled by a fuzzy logic configuration using a Fuzzy Inference System (FIS).

Keywords: Petri-nets, Fuzzy logic, Discrete systems, Robot navigation

1. INTRODUCTION

Nowadays, intelligent machines are employed in a wide range of situations addressed in modern life, due to their ability to interact dynamically and efficiently within natural environments (Katić et al., 2003). Smart humanoid robots are complex devices focused on functionality and designed to execute a set of different tasks. These devices, are prone to move safety within a specific environment, being able to collect information and process it to execute complex tasks. Navigation in natural environments is an important issue in the field of mobile robots, which consists basically in the planning and execution of movements, from a starting point in space, towards a specific target, within an environment, commonly surrounded by obstacles (Mohanta et al., 2011). In the last decades, robot navigation has been an emerging field of research where many algorithms have been tested to solve this problem, such as fuzzy logic (FL), genetic algorithms (GA), artificial neural networks (ANN) and hybrid techniques (HT); among others, see (Parhi et al., 2011). This research is focused on the implementation of a navigation strategy for a humanoid robot, so that the robot is able to move to a specific place within a restricted environment and execute a predetermined task at the end.

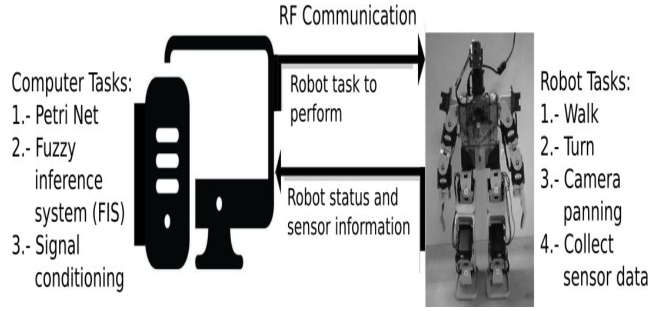

Fig. 1. Diagram of the methodology.

One of the first works related to use Petri nets with fuzzy logic was proposed in (Looney, 1988), where a new structure of a Petri net was defined, so it can deal with uncertainties by adding membership functions and fuzzy rules to places and transitions. A few years later in (Tsuji et al., 1990) an extended approach for fuzzy petri nets was proposed, the extended definition allow the analysis of the structural properties, something not possible in (Looney, 1988) model. Later in (Murata, 1996) fuzzy logic was used to add time as an element of an ordinary Petri net by defining fuzzy sets to manage temporal uncertainty.

The proposed strategy consists of a fuzzy system, viewed as a decision-making block for the navigation problem. The fuzzy system will respond to the sensorial information by executing specific actions such as to turn left or right. Additionally a Petri Net has been used to supervise and control some hierarchical robot tasks (Milutinovic et al., 2002), including activation of the fuzzy system during the exploration phase; improving robot performance (Farinelli et al., 2006).

The employed platform was a BIOLOID Premium Robot, being adapted with three infrared (IR) sensors to measure the distance between the robot and the obstacles found in the environment, during their displacement along the path to the target. The humanoid robot was also adapted with a vision processing module (VPM). This vision module is used to isolate color scenarios which are part of the environment by means of an embedded algorithm. The VPM was employed as a color detection sensor only. Information coming from the IR sensors, as well as the color detected from the VPM, was send wirelessly to a PC. The use of an external computing platform is required due to the computing limitations of the embedded robot microprocessor, improving computing performance of the fuzzy system and the Petri Nets. Unfortunately, a bottleneck related to the data flow, established between the humanoid robot and the external PC platform is unavoidable under this approach.

2. THEORETICAL FRAMEWORK

2.1 Petri Nets

Petri Nets (PN) is a formal and graphical appealing language which is appropriate for modelling systems with concurrency, characterized by being asynchronous, distributed, parallel, non-deterministic and stochastic (Murata, 1989), (David et al., 2010). PN can be used as a graphic tool in the description of this system, similar to flux and block diagrams. In PN, tokens are used to simulate dynamical and concurrent system activities. As a mathematical description tool, with PN it is possible to define a state equation, an algebraic equation and other mathematical models governing a certain system.

2.2 Fuzzy Systems

A fuzzy system can be used to solve human-like challenging problems if there is some heuristic knowledge about the solution in the form of linguistic rules of the form if-then rules. A fuzzy inference system (FIS) is a system that uses the fuzzy set theory to map inputs to outputs. A FIS is partitioned in three main blocks: Fuzzification, where crisp inputs are converted to fuzzy values by assigning them a membership value, in all the fuzzy sets defined in each input variable. It follows a fuzzy inference process, where fuzzy rules are computed. Then, a defuzzification process is defined to disambiguate the output fuzzy sets obtained as a result of computing the fuzzy rules. Fuzzy controllers are designed taking into account computing efficiency issues when they are implemented in mobile robots, due to the inherent limitations in power computing. Therefore, the use of trapezoidal shapes for the input sets, as well as fuzzy singletons for the fuzzy sets defined at the output variables is preferred. Also, fuzzy operators like MIN and MAX, are more convenient, due to their computational simplicity. Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. (Lee, 2004).

3. DEVELOPMENT

3.1 Sensors and camera

IR sensors used, are able to operate in an interval of distances ranging from 20 cm to 130 cm. To calculate the distance between the robot and a specific object, the VPM is used. The approach consists in retrieving the number of pixels found in the color region which is being searched. Before conducting this approach, a calibration phase which consists of a set of object images placed each time at different distances was obtained. An experimental representation of this relation is depicted in Fig. 2 and ecuation 1 is use to calculate the distance, where p is the number of pixels mesuare by the VPM and d is the distance to the objective in centimeters.

$$d = 905.5p^{-0.4162} - 10.96 \tag{1}$$

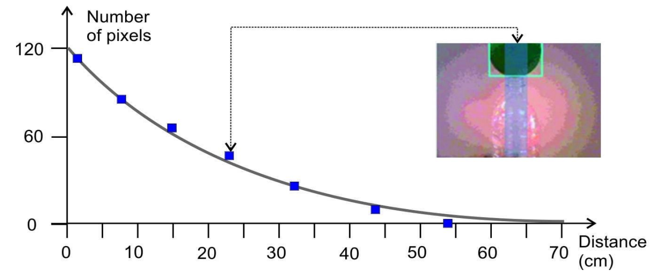

Fig. 2. Samples taken with the camera to define a relation between the number of pixels versus the distance to the object.

The robot executes the object tracking by means of panning camera movements. The fuzzy inference system is used when the robot haven't found the objective so it needs to explore the unknown environment in order to find it, while avoiding collisions with obstacles and keeping a safe distance from walls. Otherwise if the robot finds the object it begins the movements required for positioning in front of the objective. Once the robot faces the object it can start approaching it. When the robot gets close enough to the target it performs its final task, which consists of knocking down the target.

3.2 Fuzzy Inference System

The implemented FIS is in charge of the robot navigation task while the target object has not been found. By means of the environmental sensor information, the mobile robot must be able to move along the environment, avoiding obstacles, moving always in a frontal direction, and at the same time searching for a free path while it is exploring the environmental space trying to find the object. The FIS used consists of a Mamndani FIS. Three input variables were defined, corresponding with the frontal, left and right distances measured by the robot sensors. Under these conditions, fuzzy set rules were designed to conduct the robot as a wall follower. Fuzzy inference process concludes with an output command, indicating to the robot the next movement to execute, for example, a left or right turn or a short or long walk. In special situations, a 180 degrees turn can be commanded.

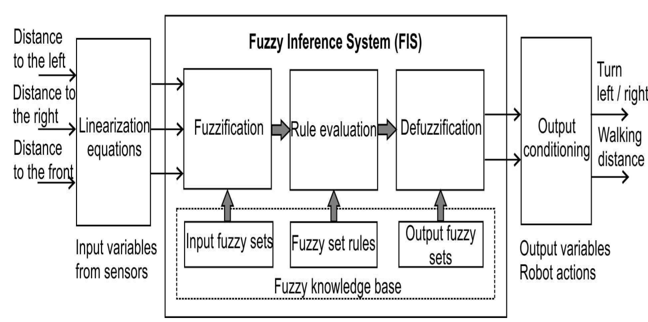

Fig. 3. Diagram of the Fuzzy Inference System used for the robot navigation.

3.3 Fuzzy Sets

Input variables, which means distance from the robot to specific lateral directions (left or right) of the FIS were partitioned in three fuzzy sets, labelled as: Closed, Medium and Far. Followed by a heuristic process, fuzzy sets were adjusted resulting in the following curved-trapezoidal-like fuzzy sets, see Fig. 4. Similarly, for the front distance sensor, four fuzzy sets were defined, see Fig. 5.

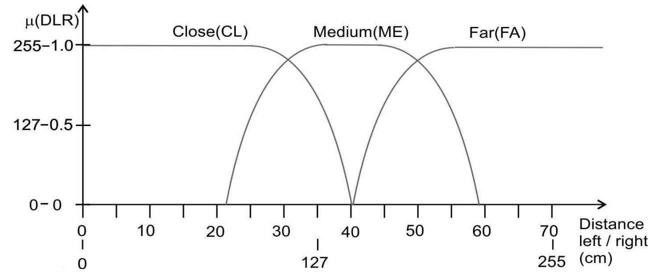

Fig. 4. Fuzzy sets define for the input signals of the lateral IR sensors.

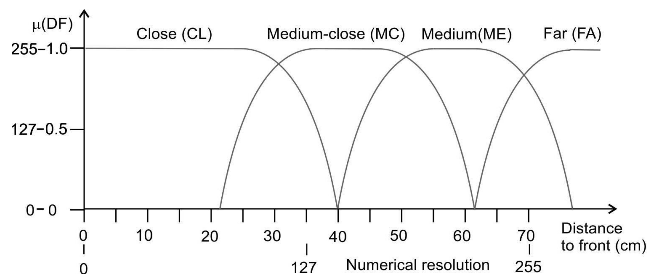

Fig. 5. Fuzzy sets defined for the input signals of the frontal IR sensor.

The output commands corresponded with three different tasks, turn right, turn left and walk forward as in Fig. 6. The action of walking, is defined in linguistic terms as a short, a medium and a long walk. By defuzzifing the output fuzzy command for the walking task, it is possible for the mobile robot to move at different distances, defined in terms of the universe of discourse, between the center of the singleton fuzzy sets named short walk, medium walk and long walk as shown in Fig. 7. It all depends on the mechanical limitations (minimum step) of the mobile robot.

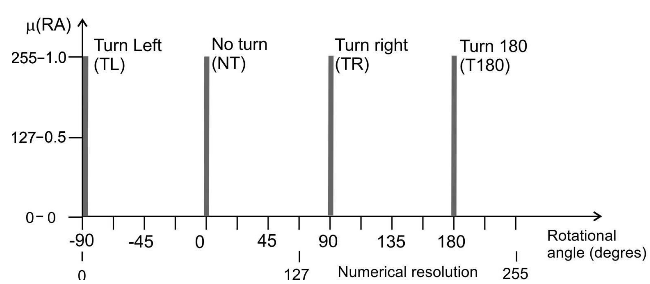

Fig. 6. Output Fuzzy sets defined as a robot task.

3.4 Fuzzy Rules

Fuzzy rules were dictated for the robot to achieve the task of walking straight, trying to follow a free path, throughout the lateral and parallel walls environment, which act as a reference for the robot orientation. The path may contain curves or corners. Some obstacles are placed inside the environment. The robot can turn in 180

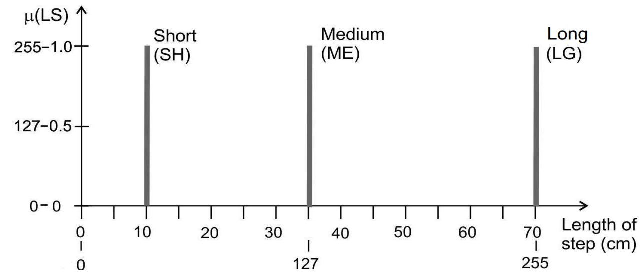

Fig. 7. Output Fuzzy sets defined as the distance walk by the robot.

degrees if it is unable to avoid some obstacle. As a result of the analysis of the fuzzy control strategy, a set of 36 fuzzy rules were defined, which are introduced in Fig. 8.

	Input variables			Output variables	
#	Distance to the front	Distance to the left	Distance to the right	Turn left / right	Walking distance
1	CL	CL	CL	T180	No action
2	FA	CL	CL	No action	LG
3	ME	CL	CL	No action	ME
4	MC	CL	CL	No action	SH
5	MC	ME	ME	No action	SH
6	MC	FA	FA	No action	SH
7	MC or CL	ME	CL	TL	No action

	Input variables			Output variables	
#	Distance to the front	Distance to the left	Distance to the right	Turn left / right	Walking distance
8	MC or CL	FA	CL	TL	No action
9	MC or CL	FA	ME	TL	No action
10	MC or CL	CL	ME	TR	No action
11	MC or CL	CL	FA	TR	No action
12	MC or CL	ME	FA	TR	No action
13	CL	ME	ME	Spe. Cas.	Spe. Cas.
14	CL	FA	FA	Spe. Cas.	Spe. Cas.

Fig. 8. Rules defined for the Fuzzy Inference System.

When present conditions suggest that the robot must realize a complete turn to avoid an obstacle, it is common to find clear space on both sides, then the last turn is adopted as a special case for the next movement. To implement the t-norms and the s-norms, MIN and MAX operators were used since a Mamndani FIS was adopted for simplicity.

3.5 Petri net

The PN that represents the supervisory control system to indicate the motions sequence that the robot has to make is shown in Fig. 9. The transitions and places are described in Table 1.

The incidence matrix is:

$$W = W^{+} - W^{-}$$

$$W = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{pmatrix} \quad (2)$$

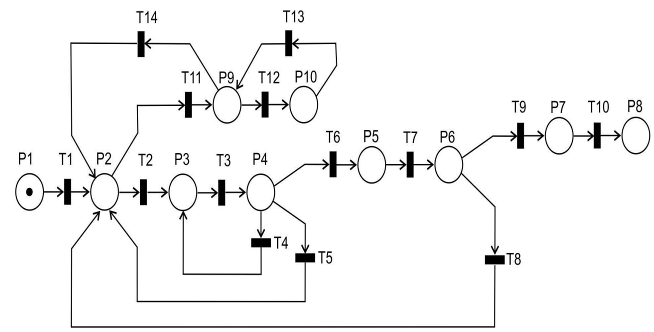

Fig. 9. Petri net designed for the robot navigation.

and its initial marking is given by $m_0^T = (1\,0\,0\,0\,0\,0\,0\,0\,0\,0)$.

Table 1. Petri net description of Places and Transitions

Places			
P_1	Robot waiting instruction	P_6	Robot walking towards goal
P_2	Robot searching goal	P_7	Robot knocking down goal
P_3	Robot turning	P_8	Robot task done
P_4	Camera focusing goal	P_9	Robot exploring (FIS)
P_5	Calculating distance to goal	P_{10}	Robot changing direction
Transitions			
T_1	Instruction recieved	T_8	Step completed
T_2	Goal found	T_9	Goal reached
T_3	Turn done	T_{10}	Goal knocked down
T_4	Goal not centered	T_{11}	Goal out of sight
T_5	Goal lost	T_{12}	FIS indicated action
T_6	Goal centered	T_{13}	New direction adquired
T_7	Robot task selected	T_{14}	New search requested

Hence the PN gets the next marking from the fundamental equation $m_k = m_i + W \cdot s$; where s is a characteristic vector and its dimension is equal to the number of places of the PN, and its subscript indicates the transition to execute, see Fig. 9. If we choose $s_1^T = (1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)$, then $(W \cdot s_1)^T = (-1\,1\,0\,0\,0\,0\,0\,0\,0\,0)$, so the new marking become $m_1^T = (0\,1\,0\,0\,0\,0\,0\,0\,0\,0)$. In the same way, if we fire the transition t_2 and t_{11}, their characteristic vectors are $s_2^T = (0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)$ and $s_{11}^T = (0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0)$ respectively. Then we obtain $(W \cdot s_2)^T = (0\,{-1}\,1\,0\,0\,0\,0\,0\,0\,0)$ and $(W \cdot s_{11})^T = (0\,{-1}\,0\,0\,0\,0\,0\,0\,1\,0)$, thus the new marking become $m_2^T = (0\,0\,1\,0\,0\,0\,0\,0\,0\,0)$ and $m_{11}^T = (0\,0\,0\,0\,0\,0\,0\,0\,1\,0)$ respectively. With these changes in the marking, it can be seen that the PN (see Fig. 9) represents all possible movements of the robot in the working space.

3.6 Properties and invariants of the Petri Net

Some behavioral and structural properties of the PN shown in Fig. 7, will be discussed in this section (Murata, 1989), (Furlán, 2013). Firstly, one of the structural properties that can be seen at a glance is about a $PN - pure$, since it has no loops. Concerning behavioral properties, this net is a $PN - ordinary$, because all the arcs have unitary weight. These properties are closely related to the fact that the network is also a $PN - safe$, since all the places are bounded to have a single token at a time. This proves that the PN is describing sequential tasks, because the robot is not able to perform them in parallel; in addition it is limited to perform them one at a time. In this sense this PN is $PN - QuasiLiveness$, inasmuch as their transitions can be fired at least once during the net evolution. This property describes how the robot has always the possibility to find the target and knock it down, as is shown in Fig. 7, with the habilitation of $T9$; namely the robot has always the possibility to finish its task in any moment. The implications of previous property is great and evidence of the power of the Petri Nets despite their simplicity. Although the above does not ensure that the robot will correctly complete all the experiments due to the PN does not model some events that can affect the robot's performance such as communication failures, bad walking, falling or if the battery runs out; but what it does assure, is that if these faults do not occur and an object is found, then the robot will find it and finish its task in a finite time, even though this can be very long. When $T10$ is fired the PN is blocked because the robot finished its tasks and it did not make sense to continue searching and exploring the environment. The PN can return to the initial marking m_1 from any other place in case $T9$ did not fire. Which means that there is always a sequence of transitions that will lead the robot to search the object.

Next the $P-$ and $T-invariants$ evaluation for the PN of Fig. 7 will be shown. In (Murata, 1989), these invariants are denoted as structural properties. We initiate with the $P - invariants$ satisfying Equation 3

$$x^T \cdot W = 0 \tag{3}$$

where the vector x is formed by the *invariants* for each place, then:

$$x^T = (x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5 \;\; x_6 \;\; x_7 \;\; x_8 \;\; x_9 \;\; x_{10}\;) \tag{4}$$

$$x^T \cdot W = \begin{pmatrix} -x_1 + x_2 \\ -x_2 + x_3 \\ -x_3 + x_4 \\ x_3 - x_4 \\ x_2 - x_4 \\ -x_4 + x_5 \\ -x_5 + x_6 \\ x_2 - x_6 \\ -x_6 + x_7 \\ -x_7 + x_8 \\ -x_2 + x_9 \\ -x_9 + x_{10} \\ x_9 - x_{10} \\ x_2 - x_9 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{5}$$

This means: $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = x_8 = x_9 = x_{10} = a$. Where a is any positive integer to meet the conditions of the $P - invariant$, because $a = c \cdot i$, where i is the unit vector formed only by ones. since all values of a are compositions of c by i, that is to say that they are linearly dependent and only the unique solution is of interest, see Eq.6; therefore the value of the invariant i is:

$$x^T = (\;1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1\;) \tag{6}$$

It can be seen that if we multiply the $P-invariant$ by any marking reachability in the PN, the result will always be 1, which tells us that there will only be one token in all network for any marking; this means that the robot can only perform one task at the same time according to its sequential behavior. The above refers to the fact that the network is conservative, because of $x > 0$ such that $x^T \cdot W = 0$, which tells us that the number of tokens is kept constant in the network; in this case there will always be a single token.

Now the $T-invariants$ have to satisfy $W \cdot y = 0$, where the vector y is formed by the *invariants* for each transition, then: then

$$W \cdot y = \begin{pmatrix} -y_1 \\ y_1 + y_2 + y_5 + y_8 - y_{11} + y_{14} \\ y_2 - y_3 + y_4 \\ y_3 - y_4 - y_5 - y_6 \\ y_6 - y_7 \\ y_7 - y_8 - y_9 \\ y_9 - y_{10} \\ y_{10} \\ y_{11} - y_{12} + y_{13} - y_{14} \\ y_{12} - y_{13} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (7)$$

where

$$y^T = (\ 0\ \ 0\ \ 1\ \ 1\ \ -1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ ) \quad (8)$$

this is not a $T-invariant$; the only solution that meets with $W \cdot y = 0$ is the trivial solution when $y = 0$, which implies that it does not have $T-invariants$. Therefore, the Petri net is not reversible. See the branch of the transition $T9$.

4. INTERFACE

An Interface was developed in C++ (see Fig. 10) to implement the FIS and PN modules. This interface executes all the communication tasks in order to retrieve information from robot sensors and sending back output commands according to the robot situation, based on the methodology explained in this paper. The interface is also used to test in real-time the robot performance, by monitoring the processed sensor information. In addition, it is possible to know the current state of the robot in the PN.

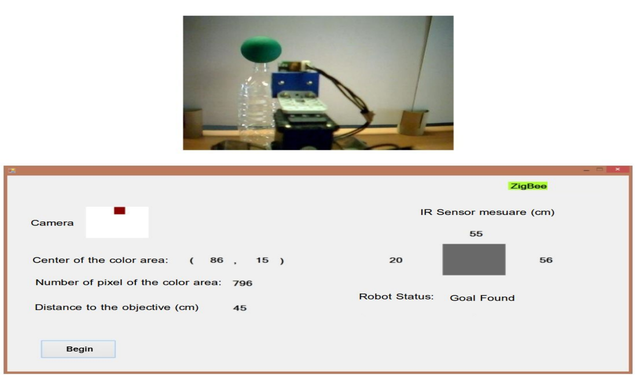

Fig. 10. Interface used to monitor robot's performance.

5. RESULTS

Several tests were conducted to evaluate the performance of the hybrid model by using different environment configurations. It was expected that the robot would move towards its objective, without colliding with the walls and obstacles distributed in its path, in all cases. Two selected environment configurations are shown as test examples. The dimensions of the work area were 2m x 1.9m. The environment configurations 1 shown in Fig. 11 and Fig. 13, were used to carry out 15 experiments.

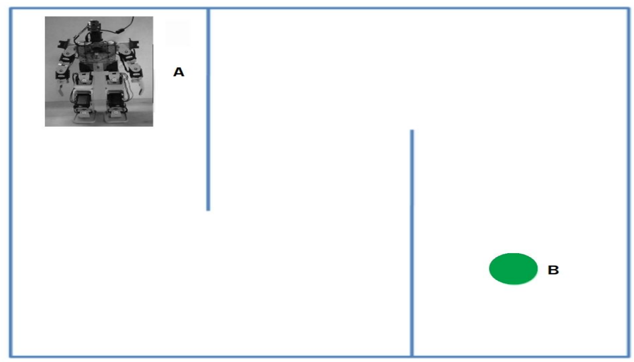

Fig. 11. Environment configuration 1, used to test the performance of the methodology.

When performing these experiments it was observed that the robot completes without problems 10 different runs. However, 3 incomplete runs were recorded due to robot failures or associated with communication problems. Some sequences of robot movements under this first configuration are presented in Fig. 12.

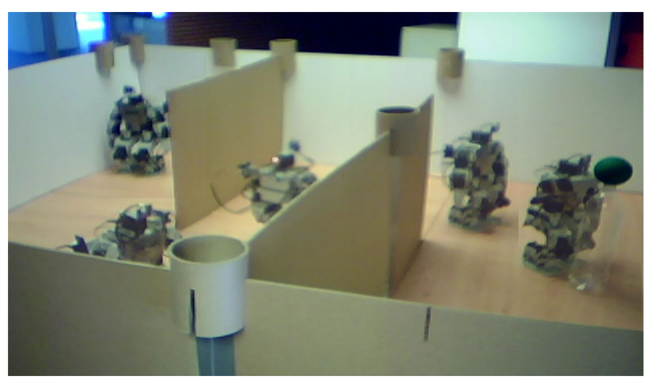

Fig. 12. Trajectory followed by the robot in environment configuration 1.

Configuration 2 is similar to that used in (Parhi et al., 2011), see Fig. 13. This configuration may served to compare the performance of the solution proposed in this work with the wheeled mobile robot found in (Parhi et al., 2011). Although both robot configurations are not the same, tests results shown similar behaviour.

In this second configuration, 10 experiments performed. Only 6 of them resulted in correct routes. As in the first configuration, the errors were mainly due to faults in the Zigbee communication. We can expect a higher rate of successful experiments if the conditions of the surface where it walks are optimal and the mechanical components have not had a significant wear.

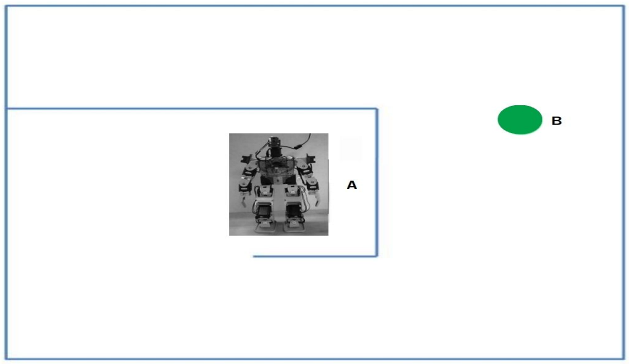

Fig. 13. Environment configuration 2, used to test the performance of the methodology.

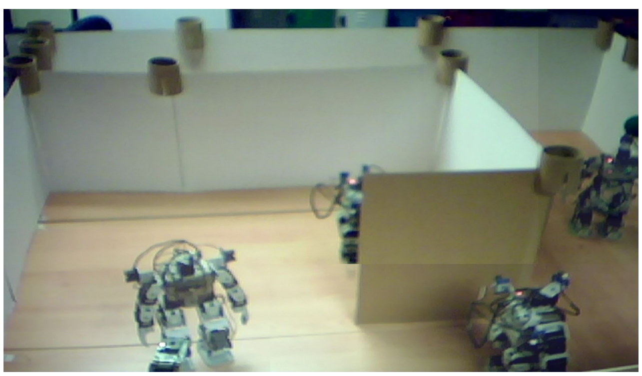

Fig. 14. Trajectory followed by the robot in environment configuration 2.

The results reported in (Parhi et al., 2011), are similar to our results are, but the great difference is a special situation for the FIS where we avoid loops in configuration 2. This methodology can be considered simpler than that shown in (Baldoni et al., 2016) since we have fewer places and transitions to achieve the same goal, that is to say obstacle avoidance. Also we explained the meaning of each place and transition of the robot behavior, nevertheless in (Baldoni et al., 2016), it is not clear what the places and transitions of the Petri net model. To view the videos of the robot operation please send an email to the authors.

6. CONCLUSIONS

A navigation method based on the sinergic combination of FL and PN was formulated and demonstrated. The hybrid model proposed in this work was found to be suitable to the platform context where it was tested, since the FIS is able to cope with the disturbances and error sensing issues, found during the walking tasks, while the PN provided the robot with the possibility to complete the task correctly for all the experiments conducted. One of the main contributions in this work is that the methodology proposed has the versatility of being able to be implemented in different mobile robots platforms, without requiring major changes, as it does not depend on its dynamic model.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided by Instituto Politécnico Nacional (IPN) and, the Secretaria de Investigación y Posgrado (SIP) for the economical support to carry out this research. This work was economically supported by SIP-IPN (grant numbers 20171548, 20170901 and 20170693), and CONACYT (grant number 155014 (Basic Research) and grant number 65 (Frontiers of Science)).

REFERENCES

Baldoni, P. D., Yang, Y., Kim, S. Y. (2016). Development of Efficient Obstacle Avoidance for a Mobile Robot Using Fuzzy Petri Nets In, *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, Pittsburgh, PA, 2016, pp. 265-269, 2016.

Furlán, F. (2013). Navegación de un robot humanoide mediante redes de Petri y lógica difusa. *Master's Thesis*, Instituto Politécnico Nacional - CIC, Mexico, 2013.

David, R., and Alla, H. (2010). *Discrete, Continuous, and Hybrid Petri Nets.*, Springer, Berlin, 2010.

Farinelli, A., Iocchi, L., Nardi, D., and Ziparo, V.A. (2006). Assignment of Dynamically Perceived Tasks by Token Passing in Multi-Robot Systems. In, *Proceedings of the IEEE, Special issue on Multi-Robot Systems*, Vol. 94, No. 7, pp. 1271–1288, 2006.

Katić, D., and Vukobratović, M. (2003). Survey of Intelligent Control Techniques for Humanoid Robots. In, *Journal of Intelligent and Robotic Systems*, Vol. 37, No. 2, pp. 117–141, 2003.

Lee, K.H. (2004). *First Course On Fuzzy Theory And Applications.*, SpringerVerlag, 2004.

Looney, C. G. (1988). Fuzzy Petri nets for rule based decisionmarking In, *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-18, 1, pp.178-183, 1988.

Milutinovic, D., and Lima, P. (2002). Petri net models of robotic tasks. In, *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Vol. 4, pp. 4059–4064, 2002.

Mohanta, J.C., Parhi, D.R., and Patel, S.K.(2011). Path Planning Strategy for Autonomous Mobile Robot Navigation Using Petri-GA Optimisation. In, *Comput. Electr. Eng.*, Vol. 37, No. 6, pp. 1058–1070, 2011.

Murata, T. (1989). Petri nets: Properties, analysis and applications In, *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541-580, 1989.

Murata, T. (1996). Temporal uncertainty and fuzzy-timing high-level Petri nets In, *Lecture Notes in Computer Science*, 1091, pp.11-28, 1996.

Parhi, D.R., and Mohanta, J.C. (2011). Navigational Control of Several Mobile Robotic Agents Using Petri-potential-fuzzy Hybrid Controller In, *Appl. Soft Comput.*, Vol. 11, No. 4, pp. 3546–3557, 2011.

Tsuji, K., Matsumoto, T. (1990). Extended Petri net models for neural networks and fuzzy inference engines - their net structural properties - In, *Proc. of IEEE ISCAS'90*, pp.2670-2673, 1990.