

# Navegación jerárquica de un robot humanoide usando redes de Petri y lógica difusa

Federico Furlán Colón, Elsa Rubio Espino, Juan Humberto Sossa Azuela,  
Víctor Hugo Ponce Ponce

Instituto Politécnico Nacional,  
Centro de Investigación en Computación, CDMX,  
México

ffurlan\_b11@sagitario.cic.ipn.mx, {erubio,hsossa,vponce}@cic.ipn.mx

**Resumen.** En este artículo se presenta un sistema de control supervisado que planea los movimientos de un robot humanoide. El sistema propuesto es una estructura de supervisión formada por dos niveles jerárquicos de un sistema a eventos discretos. La parte superior está representada por una red de Petri que se comporta como un supervisor que indica la secuencia de movimientos que el robot debe realizar. Por otro lado, el nivel inferior está representado por un robot humanoide en un ambiente controlado. Las decisiones del robot durante la fase de exploración es modelada mediante una configuración de lógica difusa que utiliza un Sistema Difuso de Inferencia (SDI).

**Palabras clave:** redes de Petri, lógica difusa, robot humanoide, navegación de robots, técnicas híbridas.

## Hierarchical Navigation of a Humanoid Robot Using Petri Nets and Fuzzy Logic

**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, humanoid robot, robot navigation, hybrid techniques.

## 1. Introducción

Hoy en día, las máquinas inteligentes son empleadas en una gran variedad de situaciones relacionadas con la vida moderna, debido a su habilidad

de interactuar dinámicamente y eficientemente dentro de ambientes comunes [4]. Los robots humanoides inteligentes son dispositivos complejos enfocados a la funcionalidad y diseñados para realizar un conjunto de tareas diversas. Estos dispositivos, son propensos a moverse de manera segura dentro de ambientes específicos, siendo capaces de obtener información y procesarla para realizar tareas complejas. La navegación en ambientes comunes es un problema importante dentro del campo de los robots móviles, la cual consiste básicamente en planear y ejecutar los movimientos, desde un punto inicial hacia un objetivo específico, dentro de un ambiente comunmente lleno de obstáculos [7].

En las últimas décadas, la navegación de robots ha sido un campo emergente de investigación donde diversos algoritmos han sido usados para resolver el problema. Estos algoritmos tienen como base algunas de las técnicas de la inteligencia artificial; como lo son la lógica difusa, los algoritmos genéticos, las redes neuronales y las técnicas híbridas; entre otras, ver [9]. La investigación de este trabajo se centra en la implementación de una estrategia de navegación en un robot humanoide, para que éste sea capaz de moverse a un lugar específico dentro de un ambiente controlado y que además realice una tarea predeterminada al finalizar su recorrido.

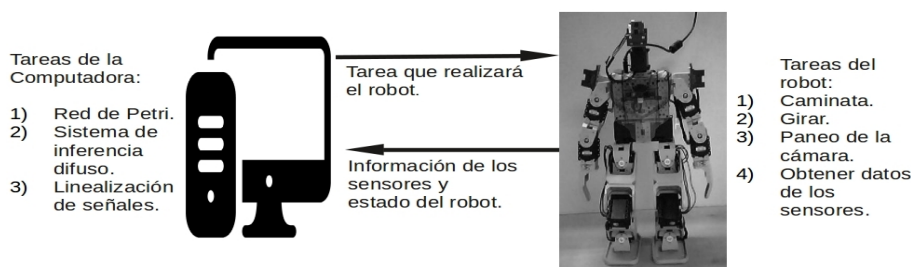


Fig. 1. Diagrama de la metodología.

La estrategia propuesta (Fig. 1) consiste de un sistema difuso, usado como un bloque de toma de decisión para el problema de navegación. El sistema difuso responde a la información de los sensores ejecutando una acción específica como puede ser girar a la izquierda o a la derecha. Adicionalmente una red de Petri es usada para supervisar y controlar algunas tareas del robot [6], incluyendo la activación del sistema difuso durante la fase de exploración; mejorando así el desempeño del robot [3,12]. La plataforma que se utilizó fue un robot BIOLOID. A este robot se le montaron tres sensores infrarrojos (IR) para medir la distancia entre él mismo y los obstáculos que pueden haber durante su desplazamiento hacia el objetivo final.

Al robot humanoide se le adaptó un módulo de visión (MV) que es usado para aislar regiones específicas de color presentes en el ambiente mediante un algoritmo embebido. El MV fue usado solamente como un sensor de detección de color. La información obtenida de los sensores infrarrojos, al igual que el

color detectado por el MV, se envían de manera inalámbrica a una computadora. El uso de una plataforma externa de cómputo es necesario, debido a las limitadas capacidades del microprocesador del robot, además de que mejora el desempeño del sistema difuso y de la red de Petri. Desafortunadamente, bajo esta metodología no se puede evitar la posibilidad de que se presente un cuello de botella relacionado con el flujo de información entre el robot humanoide y la computadora. Para probar la estrategia de control del movimiento del robot se acondicionó una superficie lisa y rígida, además se limitó el número de colores presentes en el entorno.

Este artículo está organizado de la siguiente manera, en la sección 2 se explica el trabajo realizado durante la investigación comenzando por describir, los sensores usados en el robot. A continuación se define el sistema difuso, los conjuntos difusos usados y las reglas que lo gobiernan. Finalmente se presenta el modelo de la red de Petri y se realiza un análisis de sus propiedades estructurales. En la sección 3 se muestra la interfaz desarrollada con la cual se llevaron a cabo las pruebas. En la sección 4 se discute sobre dos casos de prueba que se realizaron para verificar el desempeño de la metodología. En la sección 5 se mencionan las conclusiones obtenidas y los posibles trabajos futuros.

## **2. Desarrollo**

### **2.1. Sensores y cámara**

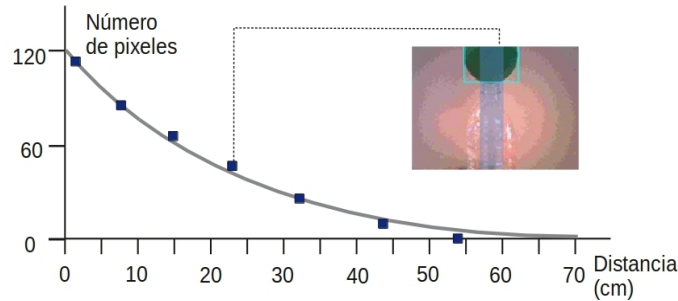
Los sensores infrarrojos usados pueden operar en un intervalo de distancias que van desde los 2 cm hasta los 130 cm. Para calcular la distancia entre el robot y el objetivo específico, se usó el MV. El enfoque usado consiste en obtener el número de píxeles que forman la región del color que está siendo buscada. Para poder implementar este método se requiere de una fase previa de calibración, la cual consiste de un conjunto de mediciones del objeto que buscará el robot colocado a diferentes distancias.

De esta manera, se puede observar que existe una relación empírica entre el número de píxeles y la distancia a la cual se encuentra el objeto de la cámara. Una representación experimental de lo anterior puede verse en la Fig. 2 de esta manera es posible aproximar un polinomio que en función del número de píxeles nos indique la distancia a la que está el objeto.

El robot realiza la búsqueda del objeto mediante movimientos de paneo del módulo de visión. En cuanto el objetivo es localizado el robot realiza una serie de movimientos para posicionarse frente a él, una vez hecho esto el robot comienza a acercarse al objetivo.

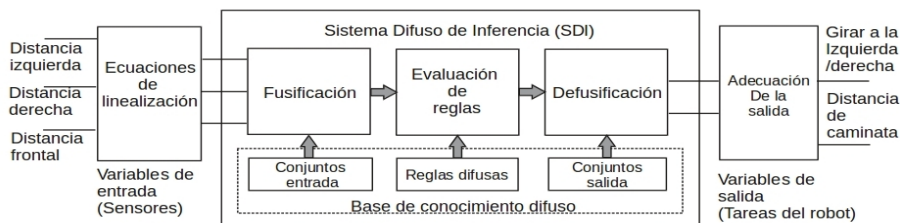
### **2.2. Sistema difuso de inferencia**

Un sistema difuso puede ser usado para resolver de manera similar a la humana problemas complejos si es que existe algún conocimiento heurístico de la solución en forma de reglas lingüísticas de la forma si-entonces. Un sistema



**Fig. 2.** Relación entre el número de píxeles y la distancia entre el objeto y la cámara.

difuso de inferencia (SDI) ver Fig. 3, es un sistema que usa la teoría de los conjuntos difusos para mapear entradas a salidas deseadas. Un SDI está dividido en tres bloques principales: Fusificación, donde las entradas son convertidas a valores difusos al asignarles un valor de membresía, para todos los conjuntos difusos definidos de cada variable de entrada. Lo siguiente es un proceso difuso de inferencia, donde las reglas difusas son evaluadas.



**Fig. 3.** Diagrama del sistema difuso de inferencia usado para la navegación del robot.

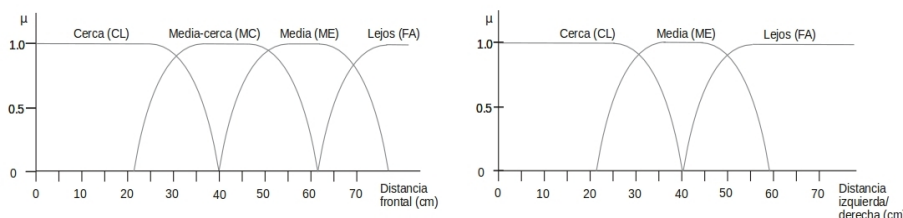
Finalmente, un proceso de defusificación es definido para asignar valores a los conjuntos difusos de salida obtenidos del cálculo de las reglas difusas. Los controladores difusos son diseñados tomando en consideración problemas con la eficiencia computacional cuando son implementados en robots móviles, debido a las inherentes limitaciones de las capacidades de cómputo. Por lo tanto es preferible el uso de conjuntos de entrada de forma trapezoidal, así como el uso de singletons para definir conjuntos difusos de salida. Además, los operadores difusos como MIN y MAX son más convenientes, dada su simplicidad computacional. Los sistemas difusos de inferencia han sido ampliamente y satisfactoriamente usados en áreas como el control automático, la clasificación de la información, el análisis de decisiones, los sistemas expertos y la visión por computadora [11,10,5].

El SDI implementado se encarga de la navegación del robot, mientras el objetivo no ha sido encontrado. En base a la información de los sensores, el

robot móvil debe ser capaz de desplazarse dentro del ambiente evadiendo los obstáculos, moviéndose siempre hacia adelante y al mismo tiempo buscando un camino despejado mientras explora el entorno tratando de localizar el objetivo. El SDI consiste en un sistema Mamdani. Se definieron tres variables de entrada, de acuerdo a las distancias frontal, izquierda y derecha obtenidas de los sensores del robot. Bajo estas condiciones, se diseñó un conjunto de reglas difusas para que el robot funcione como un seguidor de paredes. El proceso del sistema difuso concluye con un comando de salida que indica al robot el siguiente movimiento que debe realizar, como por ejemplo, un giro a la derecha o izquierda, o una caminata larga e incluso en situaciones especiales un giro de 180°.

### 2.3. Conjuntos difusos

Las variables de entrada, que significan distancia del robot en direcciones específicas (izquierda o derecha) del SDI, fueron particionadas en tres conjuntos difusos, nombrados como: Cerca, Media y Lejos. Usando un proceso heurístico, se ajustaron los conjuntos difusos dando como resultado las formas curvo-trapezoidal de los conjuntos difusos, ver Fig. 4. De manera similar, para el sensor frontal, cuatro conjuntos difusos fueron definidos.



**Fig. 4.** Conjuntos difusos definidos para las señales de entrada de los sensores infrarrojos.

Las instrucciones de salida corresponden a tres diferentes tareas, giro a la derecha, giro a la izquierda y caminata; como se puede ver en la Fig. 5. La acción de caminata, esta definida por los términos lingüísticos Corta, Media y Larga. Al defusificar los comandos de salida de la tarea caminata, es posible que el movimiento del robot sea de diferentes distancias, definidas por los términos del universo de discurso, entre los centros de los conjuntos difusos singleton nombrados caminata corta, caminata media y caminata larga como se muestra en la Fig. 5. Lo anterior depende de las limitaciones mecánicas (avance mínimo) del robot móvil.

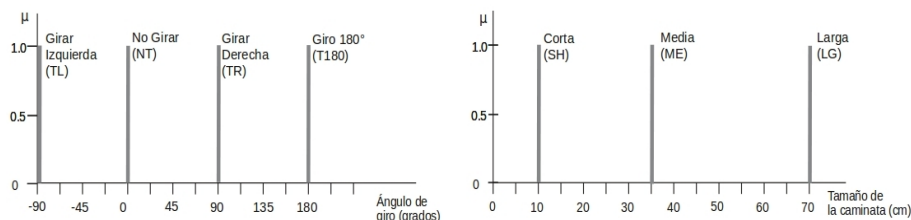


Fig. 5. Conjuntos difusos de salida definidos como acciones del robot.

### 2.4. Reglas difusas

Las reglas difusas fueron definidas para que el robot realice la tarea de caminar hacia adelante, tratando de seguir un camino libre de obstáculos, dentro del entorno de pruebas, compuesto por paredes rectas, las cuales sirven como referencia para la orientación del robot. El robot puede girar 180° si no le es posible evadir algún obstáculo o en caso de que no exista camino para seguir desplazándose hacia adelante. Como resultado del análisis de la estrategia del sistema difuso, se definió un conjunto de 36 reglas difusas, las cuales se presentan en la Fig. 6. Es posible que se presenten situaciones en las cuales el robot necesite realizar un giro para evadir un obstáculo, pero es común que el espacio libre en ambos lados haga viable girar hacia la derecha o izquierda. En este caso, el robot lo considerará un caso especial, éste último referenciado en la Fig. 6 como: *Caso Esp.* Si es así, el robot adoptará la dirección del último giro que ha realizado para continuar su desplazamiento. Para implementar la norma-t y la norma-s, los operadores MIN y MAX fueron usados ya que se empleó un sistema Mamdani por su simplicidad.

#	Variables de entrada			Variables de salida		#	Variables de entrada			Variables de salida	
	Distancia frontal	Distancia izquierda	Distancia derecha	Giro izq / der	Distancia caminata		Distancia frontal	Distancia izquierda	Distancia derecha	Giro izq / der	Distancia caminata
1	CL	CL	CL	T180	----	8	MC o CL	FA	CL	TL	----
2	FA	CL	CL	----	LG	9	MC o CL	FA	ME	TL	----
3	ME	CL	CL	----	ME	10	MC o CL	CL	ME	TR	----
4	MC	CL	CL	----	SH	11	MC o CL	CL	FA	TR	----
5	MC	ME	ME	----	SH	12	MC o CL	ME	FA	TR	----
6	MC	FA	FA	----	SH	13	CL	ME	ME	Caso Esp.	Caso Esp.
7	MC o CL	ME	CL	TL	----	14	CL	FA	FA	Caso Esp.	Caso Esp.

Fig. 6. Reglas definidas para el sistema difuso de inferencia.

### 2.5. Red de Petri

Las redes de Petri (RdP) son un interesante lenguaje gráfico y formal el cual es apropiado para el modelado de sistemas con concurrencias, caracterizados por ser asíncronos, distribuidos, paralelos, no deterministas y estocásticos [2,1,8]. Las

RdP pueden ser usadas como una herramienta gráfica en la descripción de un sistema, similar a los diagramas de flujo o de bloques. En las RdP, se utilizan tokens para simular actividades dinámicas y concurrentes del sistema. También pueden ser usadas como una herramienta de descripción matemática, ya que es posible definir una ecuación de estado o ecuación fundamental, una ecuación algebraica de la secuencia de eventos y otros modelos matemáticos que gobiernan a los sistemas.

La RdP que representa el control supervisor del sistema, indica la secuencia de movimientos que el robot debe realizar. Ver la Fig. 7.

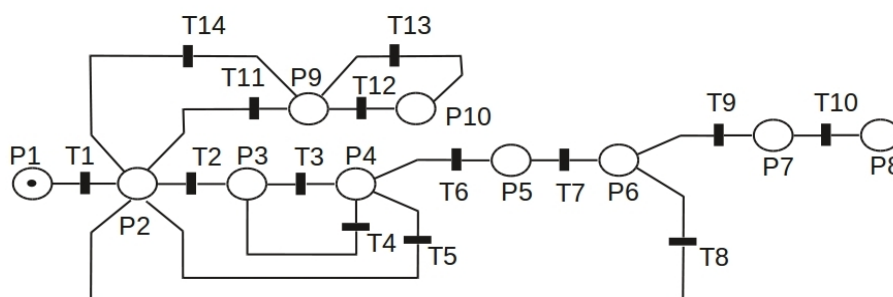


Fig. 7. Red de Petri.

La descripción de las transiciones y lugares de la RdP se describen en la Tabla 1.

Tabla 1. Descripción de lugares y transiciones de la RdP.

Lugares			
$P_1$	Robot esperando instrucción	$P_6$	Robot caminando al objetivo
$P_2$	Robot buscando objetivo	$P_7$	Robot derribando el objetivo
$P_3$	Robot girando	$P_8$	Tarea del robot completada
$P_4$	Robot alineado con objetivo	$P_9$	Robot explorando
$P_5$	Calculando distancia al objetivo	$P_{10}$	Robot cambiando dirección

Transiciones			
$T_1$	Instrucción recibida	$T_8$	Caminata completada
$T_2$	Objetivo encontrado	$T_9$	Objetivo alcanzado
$T_3$	Giro realizado	$T_{10}$	Objetivo derribado
$T_4$	Objetivo no alineado	$T_{11}$	Objetivo no detectado
$T_5$	Objetivo perdido	$T_{12}$	Acción indicada por el SDI
$T_6$	Objetivo centrado	$T_{13}$	Nueva dirección fijada
$T_7$	Tarea del robot seleccionada	$T_{14}$	Nueva búsqueda requerida

La matriz de incidencia es:

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

y su marcaje inicial está dado por:  $m_0 = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ .

Entonces la RdP obtiene su marcaje siguiente de la ecuación fundamental:

$$m_k = m_i + W \cdot s, \quad (2)$$

donde  $s$  es el vector característico y su dimensión es igual al número de lugares de la RdP, y su subíndice indica la transición disparada, ver Fig. 7.

Si elegimos  $s_1^T = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ , entonces  $(W \cdot s)^T = (-1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ , tal que el nuevo marcaje es  $m_1 = (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$ . De la misma manera, si disparamos las transiciones  $t_2$  y  $t_{11}$ , sus vectores característicos son  $s_2^T = (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$  y  $s_{11}^T = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0)$ . Se obtiene ahora  $(W \cdot s_2)^T = (0\ -1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$  y  $(W \cdot s_{11})^T = (0\ -1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)$ , por lo tanto el nuevo marcaje es  $m_2 = (0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$  y  $m_{11} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)$  respectivamente. Con estos cambios en el marcaje de la RdP, puede verse que la RdP de la Fig. 7, representa todos los posibles movimientos del robot en el espacio de trabajo.

## 2.6. Propiedades de la red de Petri

En esta sección, serán analizadas algunas propiedades estructurales y de comportamiento de la RdP mostrada en la Fig. 7. Primeramente, una de las principales propiedades estructurales que se observan, es que la RdP es *pura*, debido a que no tiene ciclos.

En relación a las propiedades del comportamiento, esta RdP es *ordinaria*, porque todos los arcos tienen pesos unitarios. Estas propiedades están estrechamente relacionadas al hecho de que la RdP también es *segura*, debido a que todos sus lugares están limitados a tener solo un token a la vez. Esto prueba que la RdP esta describiendo tareas secuenciales, porque el robot no es capaz de realizarlas en paralelo; adicionalmente está limitado a realizarlas una a la



vez. En este sentido, la RdP es *quasiviva*, ya que la mayoría de sus transiciones pueden ser disparadas por lo menos una vez durante la evolución de la red.

Esta propiedad describe cómo el robot siempre tiene la posibilidad de encontrar el objetivo y derribarlo; tal y como se muestra en la Fig. 7. En esta misma figura, se puede ver que la transición  $T_9$ , se utiliza para darle al robot la posibilidad de finalizar su tarea en cualquier momento. Las implicaciones de la propiedad anterior son importantes, y hace evidente el poder de descripción que tienen las redes de Petri, a pesar de la sencillez de su uso. Hay que dejar en claro que aunque lo anterior no asegura que el robot completará de manera correcta todos los experimentos, debido a que esta RdP no modela algunos de los eventos que pueden afectar el desempeño del robot como pueden ser las fallas en la comunicación inalámbrica con la computadora, las fallas que pueden ocurrir por una inadecuada marcha del robot o que la batería se agote, sí se asegura que en caso de que no ocurra una falla no modelada y que exista un objetivo dentro del área de trabajo, entonces el robot encontrará y terminará su tarea en un tiempo finito, aunque este tiempo pueda ser muy grande.

Por otro lado, cuando  $T_{10}$  es disparada, la RdP se bloquea porque el robot ha finalizado su tarea y no tiene sentido que siga buscando el objetivo o explorando su entorno. Para finalizar, se debe resaltar que esta RdP, puede regresar al marcaje inicial  $M_1$  desde cualquier lugar mientras no se haya disparado la transición  $T_9$ . Lo cual significa que siempre existe una secuencia de transiciones que llevarán al robot a seguir buscando el objetivo. A continuación se mostrará el análisis de los P-invariantes y los T-invariantes de la RdP. En [8], estos invariantes son denotados como propiedades estructurales. Se iniciará con los P-invariantes, por lo que es necesario resolver la ecuación siguiente:

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

donde

$$x^T \cdot W = \begin{pmatrix} -x_1 + x_2 \\ -x_2 + x_3 \\ -x_3 + x_4 \\ x_3 - x_4 \\ x_2 - x_4 \\ -x_4 + x_5 \\ -x_5 + x_6 \\ x_2 - x_6 \\ -x_6 + x_7 \\ -x_7 + x_8 \\ -x_2 + x_9 \\ -x_9 + x_{10} \\ x_9 - x_{10} \\ x_2 - x_9 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{4}$$

lo cual significa que:

$$x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = x_8 = x_9 = x_{10} = a,$$

donde  $a$  es cualquier entero positivo que satisfaga las condiciones de los P-invariantes, debido a que  $a = c \cdot i$ , donde  $i$  es el vector unitario formado solo por unos, debido a que todos los valores de  $a$  son composiciones de  $c \cdot i$ , lo que nos indica que son linealmente dependientes y que solo la solución única es de interés; por lo tanto la solución de los P-invariantes es  $i$ :

$$x^T = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1).$$

Puede verse que si multiplicamos los P-invariantes por cualquier marcaje alcanzable de la RdP, el resultado siempre será 1, lo cual nos indica que solo existe un token en toda la red para cualquier marcaje; lo cual significa que el robot solo puede realizar una tarea a la vez lo cual es acorde con su comportamiento secuencial. Lo anterior se refiere al hecho de que la red es *conservativa*, porque  $x > 0$  tal que  $x^T \cdot W = 0$ , lo cual indica que el número de tokens se mantiene constante en la red; en este caso siempre habra un solo token.

Ahora los T – invariantes deben satisfacer:

$$W \cdot y = 0, \tag{5}$$

entonces:

$$W \cdot y = \begin{pmatrix} -y_1 \\ y_1 + y_2 + y_5 + y_8 - y_{11} + y_{14} \\ y_2 - y_3 + y_4 \\ y_3 - y_4 - y_5 - y_6 \\ y_6 - y_7 \\ y_7 - y_8 - y_9 \\ y_9 - y_{10} \\ y_{10} \\ y_{11} - y_{12} + y_{13} - y_{14} \\ y_{12} - y_{13} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{6}$$

donde:

$$y^T = (0\ 0\ 1\ 1\ -1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 1).$$

El cual no es un T – invariante; la única solución que cumple con  $y^T \cdot W = 0$  es la solución trivial cuando  $y = 0$ , lo cual implica que la RdP no tiene T-invariantes. Por lo tanto, la red de Petri no es *reversible*. Lo cual se puede ver debido a la existencia de la transición  $T_9$ .

### 2.7. Interfaz

Se desarrolló una interfaz computacional en C++ (ver Fig. 8) para implementar los módulos del SDI y la RdP. La interfaz realiza las tareas de comunicación, con el objetivo de obtener la información de los sensores del robot y enviar al robot los comandos de salida de acuerdo con la situación en la cual se encuentra, basado en la metodología previamente explicada. La interfaz también funciona para monitorear el desempeño del robot en tiempo real, al monitorear la información de los sensores y del módulo de visión. Adicionalmente es posible conocer el estado actual del robot con relación a la RdP.

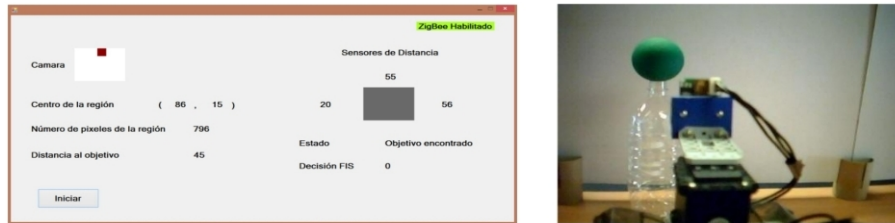


Fig. 8. Interfaz desarrollada para monitorear el desempeño del robot.

### 3. Resultados

Se realizaron varias pruebas para evaluar el desempeño del modelo híbrido propuesto, se usaron diferentes configuraciones del ambiente de trabajo. El comportamiento esperado es que el robot se mueva hacia el objetivo, sin chocar con las paredes u obstáculos que puedan obstruir su camino, ésto se espera para todas las configuraciones. Se seleccionaron dos configuraciones para ser analizadas en este artículo, pero si se quieren consultar más pruebas y configuraciones, se puede consultar [1], ahí se encontrarán más pruebas y con mayor detalle. La configuración 1 de los ambientes de prueba, es mostrada en la Fig. 9, con este escenario, se llevaron a cabo 15 experimentos. Durante las pruebas se observó que el robot completó 12 recorridos sin ningun problema y que los 3 recorridos incompletos pero fueron debidos a fallas en la comunicación con la computadora. Una secuencia del movimiento del robot en este ambiente de pruebas es presentado en la Fig. 10.

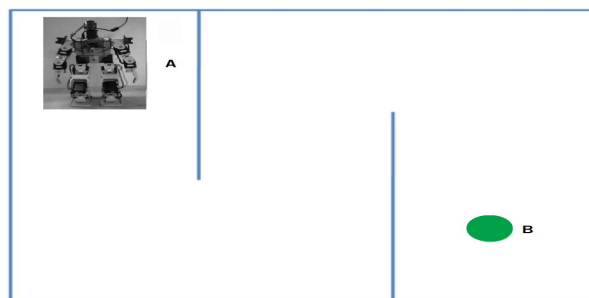


Fig. 9. Configuración 1, usada para probar el desempeño de la metodología propuesta.

La configuración 2 es similar a la usada en [9], ver Fig. 11. Esta configuración puede servir para comparar el desempeño de la solución propuesta en ambos trabajos. Aunque los tipos de robots son distintos, debido a que uno es un robot humanoide y el otro un robot con ruedas, los resultados muestran un comportamiento similar en ambos casos.

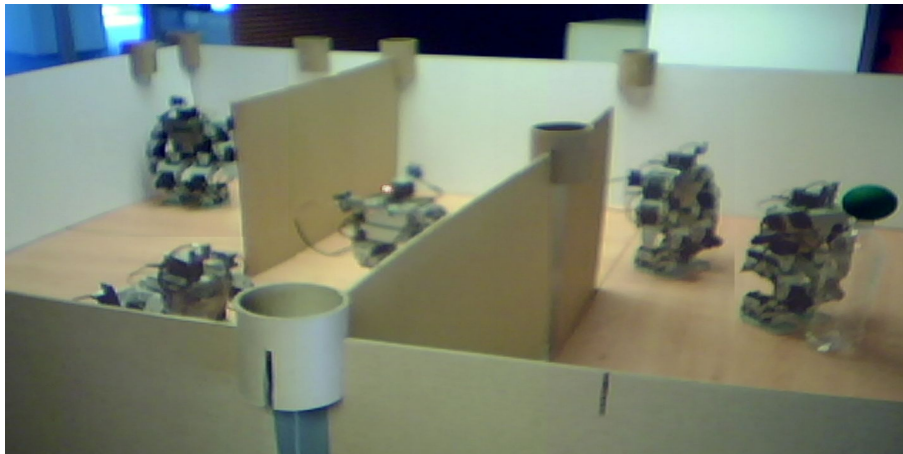


Fig. 10. Trayectoria seguida por el robot durante las pruebas en la configuración 1.

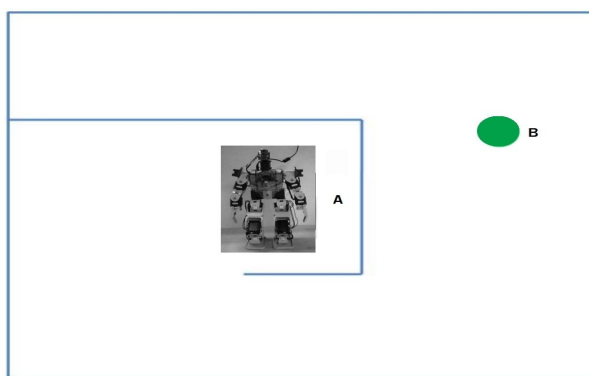


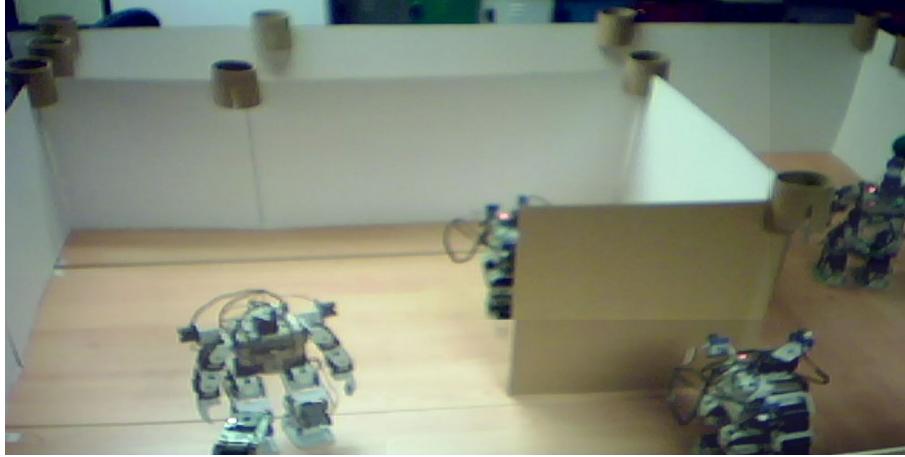
Fig. 11. Configuración 2, usada para probar el desempeño de la metodología propuesta.

En esta segunda configuración, ver Fig. 12, se realizaron 10 experimentos. De los cuales 8 fueron recorridos completos. Como en el caso de la primera configuración. Las dimensiones del área de trabajo fueron  $2\text{m} \times 1.9\text{m}$ .

#### 4. Conclusiones

Se propuso e implementó un método de navegación basado en la combinación de la lógica difusa y las redes de Petri. Esto representa una propuesta de un control supervisado, donde la RdP funge como supervisor y el robot como un dispositivo supervisado.

El uso de la lógica difusa, permite darle la oportunidad al robot de mejorar su desempeño al caminar; ya que el robot se va aproximando al objetivo de una



**Fig. 12.** Trayectoria seguida por el robot durante las pruebas en la configuración 2.

manera suave y gradual para disminuir las posibilidades de choque y/o de derribo involuntario.

Finalmente este uso combinado de dos técnicas de inteligencia artificial permite describir al sistema en conjunto, como un modelo híbrido con jerarquía, además de que como puede observarse en los experimentos, mejora el desempeño del robot al realizar las tareas recibidas a través de la computadora.

Resumiendo, el modelo híbrido propuesto en este trabajo fue adecuado para la plataforma en la cual se probó, ya que el SDI fue capaz de tolerar los problemas causados por las perturbaciones de la marcha del robot y los errores de sensado, mientras que la RdP proporciona al robot la posibilidad de completar su tarea de manera correcta para todos los experimentos realizados. La metodología propuesta cuenta con la versatilidad de poder ser implementada en diferentes plataformas de robots móviles, sin necesitar de cambios mayores ya que no depende del modelo dinámico del robot.

**Agradecimientos.** Los autores agradecen el apoyo recibido por el Instituto Politécnico Nacional (IPN) y la Secretaría de Investigación y Posgrado (SIP), por el apoyo económico brindado para llevar a cabo esta investigación. Así mismo, se agradece el apoyo económico a la SIP-IPN (números 20171548, 20170901 y 20170693), y CONACYT (número 155014 (Investigación Básica) y número 65 (Fronteras de la Ciencia)). Federico Furlán agradece al CONACYT por la beca otorgada para la realización de sus estudios de posgrado, así como el apoyo del programa BEIFI-IPN.

## Referencias

1. Furlán, F.: Navegación de un robot humanoide mediante redes de Petri y lógica difusa. Instituto Politécnico Nacional, CIC, Tesis de Maestría (2013)
2. David, R., Alla, H.: Discrete, Continuous, and Hybrid Petri Nets. Springer, Berlin (2010)
3. Farinelli, A., Iocchi, L., Nardi, D., Ziparo, V. A.: Assignment of Dynamically Perceived Tasks by Token Passing in Multi-Robot Systems. Proceedings of the IEEE, Special issue on Multi-Robot Systems, Vol. 94, No. 7, pp. 1271–1288 (2006)
4. Katić, D., Vukobratović, M.: Survey of Intelligent Control Techniques for Humanoid Robots. Journal of Intelligent and Robotic Systems, Vol. 37, No. 2, pp. 117–141 (2003)
5. Lee, Kwang Hyung: First Course On Fuzzy Theory And Applications. Springer Verlag (2004)
6. Milutinovic, D., Lima, P.: 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)
7. Mohanta, J. Ch., Parhi, D. R., Patel, S.K.: Path Planning Strategy for Autonomous Mobile Robot Navigation Using Petri-GA Optimisation. Comput. Electr. Eng., Vol. 37, No. 6, pp. 1058–1070 (2011)
8. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE, Vol. 77, No. 4, pp. 541–580 (1989)
9. Parhi, D.R., Mohanta, J.Ch.: Navigational Control of Several Mobile Robotic Agents Using Petri-potential-fuzzy Hybrid Controller. Appl. Soft Comput., Vol. 11, No. 4, pp. 3546–3557 (2011)
10. Perera, L. P., Carvalho, J. P., Guedes Soares, C.: Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. Journal of Marine Science and Technology, Vol. 16, No. 1, pp. 84–99 (2011)
11. Terano, T., Asai, K., Sugeno, M.: Fuzzy Systems Theory and Its Applications. Academic Press Professional, Inc. (1992)
12. Wai, Rong-Jong, Liu, Chia-Ming, Lin, You-Wei: Robust path tracking control of mobile robot via dynamic petri recurrent fuzzy neural network. Soft Computing, Vol. 15, No. 4, pp. 743–767 (2010)