

# DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE OBSTÁCULOS PARA ROBOTS AUTÓNOMOS BASADO EN LÓGICA BORROSA

Daniel Vicente Rodrigo Muñoz  
Universidad Complutense de Madrid, danielvr@ucm.es

Matilde Santos  
Universidad Complutense de Madrid, msantos@ucm.es

## Resumen

*En el presente trabajo se propone el diseño y la implementación de un sistema de detección de obstáculos para un robot móvil. El sistema de control utiliza la información proporcionada por sensores de ultrasonidos para la navegación. Se han diseñado dos controladores basados en lógica borrosa, de diferente nivel de complejidad. Se han implementado en una plataforma de desarrollo de Arduino, con el software proporcionado por MATLAB y Simulink, y se ha probado en un sistema real. Los resultados de los experimentos con el robot han probado que el control borroso permite al robot seguir una trayectoria evitando los obstáculos incluso en escenarios relativamente complejos.*

**Palabras Clave:** Control inteligente, lógica borrosa, robot móvil, detección de obstáculos, navegación.

## 1 INTRODUCCIÓN

El potencial de aplicación de los robots móviles está estrechamente relacionado con el nivel de autonomía que presenten, pudiendo realizar una serie de tareas en entornos desconocidos, sin la necesidad de un guiado constante.

Existe una gran variedad de robots totalmente autónomos. Estos deben de ser capaces de desplazarse sin asistencia humana, a partir de información sobre el entorno, evitando situaciones perjudiciales tanto para ellos como para otros objetos.

Uno de los principales problemas de la navegación autónoma de los robots móviles es la detección de obstáculos [9]. En este trabajo se propone un sistema de lógica borrosa que evalúa la información proporcionada por sensores de ultrasonidos de un robot móvil para guiarle en la navegación a través de obstáculos.

El sistema se ha implementado en una plataforma real, realizando experimentos en diversos escenarios de forma satisfactoria.

## 2 DESCRIPCIÓN DEL SISTEMA

La navegación se consigue, en un robot móvil de dos ruedas (Figura 1), mediante cambios de velocidad. El modelo que se va a utilizar para su simulación viene dado por las expresiones de posición y orientación siguientes:

$$\dot{x} = V \cos \theta \quad (1)$$

$$\dot{y} = V \sin \theta \quad (2)$$

$$\dot{\theta} = W \quad (3)$$

Donde  $V$  es la velocidad lineal promedio de las velocidades de las dos ruedas, y  $W$  la velocidad angular.

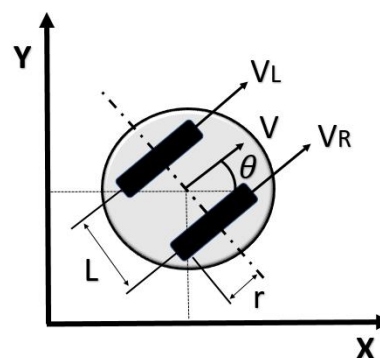


Figura 1: Esquema cinemático del robot móvil

Hay que tener en cuenta una serie de hipótesis para generalizar el comportamiento del robot [3]:

- El robot se desplaza en una superficie plana idealmente sin rozamiento.
- Los ejes de las ruedas son perpendiculares al suelo por donde se desplaza.
- Se mueve únicamente por la fuerza ejercida por el movimiento rotacional de las ruedas.

- Es un mecanismo sólido rígido y sin partes flexibles.

## 2.1 ARQUITECTURA DEL ROBOT

En la Figura 2 se puede contemplar la arquitectura propuesta para el robot.

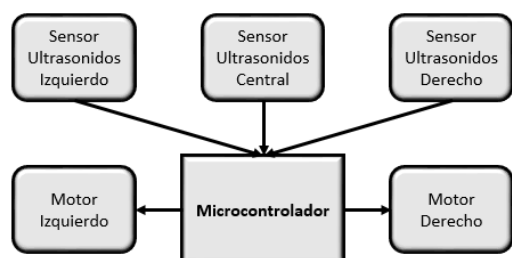


Figura 2: Arquitectura hardware del robot [11]

El sistema recopila información del mundo exterior a través de unos sensores de ultrasonidos. Se ha usado el sensor de proximidad HC-SR04, que utiliza un sonar para medir distancias. Puede detectar distancias desde los 2 cm hasta los 450 cm con una precisión de 0.3 cm. Para esta aplicación se ha limitado el rango hasta los 100 cm.

Los datos recogidos son tratados en el microcontrolador, que contiene el controlador basado en lógica borrosa, y permite actuar sobre los motores que determinan la velocidad de cada rueda. El microcontrolador utilizado para este sistema es Arduino Uno, basado en el microcontrolador ATmega328. La placa de desarrollo consta de catorce pines E/S digitales y seis pines E/S analógicos, con una memoria flash de 32 kB y una memoria EEPROM de 1 kB, con una frecuencia de 16 MHz.

Se dispone de dos motores DC, uno para cada rueda, cuya velocidad media va desde 170 rpm a 250 rpm. Para controlar los motores DC desde la placa del microcontrolador se emplea un módulo del circuito integrado L298N, con el fin de entregar más corriente a los motores, alimentándolos mediante una batería externa. Para controlar el sentido del motor se generan dos señales PWM; una señal será la inversa de la otra, y en función del ciclo de trabajo de cada una se establecerá una velocidad y un sentido.

## 2.2 IMPLEMENTACIÓN

La implementación del robot ha requerido de una serie de módulos específicos que ha permitido gestionar la información recibida de los sensores, la información enviada a los motores, y la correcta alimentación de todos los elementos que forman parte del sistema.

La Tabla 1 muestra la asignación de señales a los diferentes pines de la placa Arduino Uno. Todos los componentes se conectan a esta placa a través del módulo para sensores, y del módulo para motores, mencionados anteriormente.

Tabla 1: Asignación de pines.

Señal	Componente	Pin
echo	Ultrasonidos Derecho	5
trigger	Ultrasonidos Derecho	4
echo	Ultrasonidos Central	3
trigger	Ultrasonidos Central	2
echo	Ultrasonidos Izquierdo	7
trigger	Ultrasonidos Izquierdo	8
IN1	Motor Derecho	6
IN2	Motor Derecho	9
IN3	Motor Izquierdo	11
IN4	Motor Izquierdo	10

Una vez evaluada la electrónica del sistema hay que ubicarla dentro de la estructura del robot móvil de configuración diferencial. En la Figura 3 se muestra la solución adoptada para este proyecto.

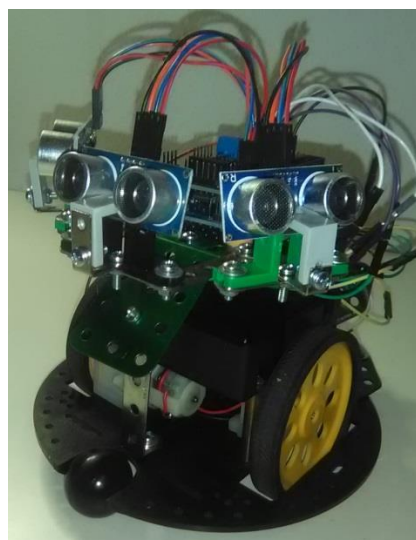


Figura 3: Solución adoptada para el robot autónomo

## 3 DISEÑO DE CONTROLADORES FUZZY PARA LA NAVEGACIÓN DE ROBOT MOVIL

El controlador basado en lógica fuzzy será el encargado de evaluar la información obtenida del entorno a través de los sensores y generar las señales adecuadas para guiar al robot a través de los obstáculos que se presenten [4, 10]. En este trabajo se han diseñado dos controladores borrosos de distinta complejidad.

Las entradas del controlador son la distancia obtenida por cada uno de los tres sensores de ultrasonidos a un obstáculo. Se han definido tres conjuntos borrosos, cerca, media y lejos, con funciones de pertenencia trapezoidales (Figura 4) en el intervalo 0-100 cm.

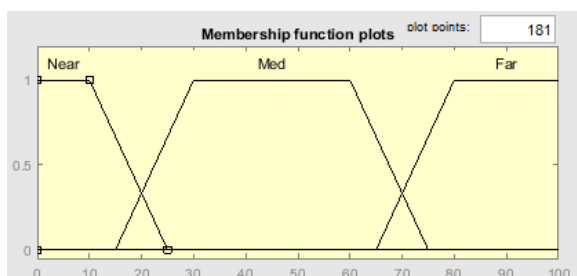


Figura 4: Entrada borrosa (distancia)

Típicamente las soluciones adoptadas por un sistema de control borroso de la velocidad de los motores se centran en el error [1, 8]. Nuestra propuesta es controlar el nivel de velocidad de cada motor en términos lingüísticos tales como rápido, o lento, adelante o atrás [11]. El controlador diseñado determina la velocidad del motor y el sentido de giro considerando los conjuntos borrosos que representan avance lento o rápido, o un retroceso lento, o rápido (Figura 5).

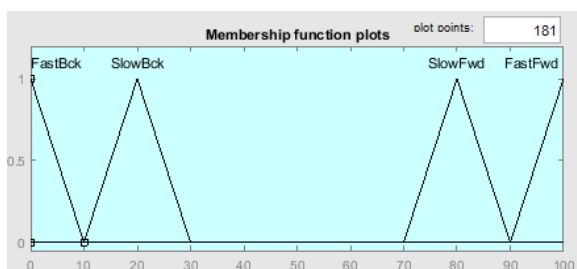


Figura 5: Salida borrosa del controlador (motor)

La salida del controlador no es un valor de velocidad del motor, sino que se corresponde con el ciclo de trabajo que debe presentar el motor para satisfacer la acción deseada. El ciclo de trabajo define las señales PWM necesarias para el control de los motores.

Para no comprometer el correcto funcionamiento de los motores y garantizar el desplazamiento del robot autónomo se ha definido que la variación del ciclo de trabajo sea como máximo de un 30%.

Se ha implementado un sistema de inferencia para reglas del tipo Mamdani y una defuzzificación de tipo centroide para obtener la salida analítica.

La Tabla 2 muestra las reglas que se aplican para definir el sentido y la velocidad de giro para cada motor, en función de la cercanía con el obstáculo detectado.

Tabla 2: Reglas borrosas del control de los motores.

Regla	Sensor Izq	Sensor Cen	Sensor Dch	Motor Izq	Motor Dch
1	Cerca	Cerca	Cerca	Retro Rápido	Retro Rápido
2	Cerca	Cerca	Media	Avance Rápido	Retro Rápido
3	Cerca	Cerca	Lejos	Avance Rápido	Retro Rápido
4	Cerca	Media	Cerca	Avance Lento	Avance Lento
5	Cerca	Media	Media	Avance Rápido	Retro Lento
6	Cerca	Media	Lejos	Avance Rápido	Retro Lento
7	Cerca	Lejos	Cerca	Avance Rápido	Avance Rápido
8	Cerca	Lejos	Media	Avance Rápido	Retro Lento
9	Cerca	Lejos	Lejos	Avance Rápido	Avance Lento
10	Media	Cerca	Cerca	Retro Rápido	Avance Rápido
11	Media	Cerca	Media	Avance Rápido	Retro Rápido
12	Media	Cerca	Lejos	Avance Rápido	Retro Rápido
13	Media	Media	Cerca	Retro Lento	Avance Rápido
14	Media	Media	Media	Avance Lento	Avance Lento
15	Media	Media	Lejos	Avance Rápido	Avance Lento
16	Media	Lejos	Cerca	Retro Lento	Avance Rápido
17	Media	Lejos	Media	Avance Rápido	Avance Rápido
18	Media	Lejos	Lejos	Avance Rápido	Avance Lento
19	Lejos	Cerca	Cerca	Retro Rápido	Avance Rápido
20	Lejos	Cerca	Media	Retro Rápido	Avance Rápido
21	Lejos	Cerca	Lejos	Avance Rápido	Retro Rápido
22	Lejos	Media	Cerca	Retro Lento	Avance Rápido
23	Lejos	Media	Media	Avance Lento	Avance Rápido
24	Lejos	Media	Lejos	Avance Rápido	Avance Lento
25	Lejos	Lejos	Cerca	Avance Lento	Avance Rápido
26	Lejos	Lejos	Media	Avance Lento	Avance Rápido
27	Lejos	Lejos	Lejos	Avance Rápido	Avance Rápido

Para definir estas reglas se ha analizado el listado propuesto en [5], obtenido de manera heurística, y se le han aplicado ciertas modificaciones para poderlas aplicar al sistema propuesto.

El controlador implementado busca la ruta más favorable, aquella más alejada de cualquier obstáculo, y que garantiza la seguridad suficiente para continuar con su camino, la que le permita alejarse de los obstáculos más próximos y le asegure que no habrá nuevos impedimentos en las proximidades.

En la Figura 6 se muestra, a modo de ejemplo, la relación entre la salida que afecta al motor derecho, a partir de la distancia obtenida por cada sensor.

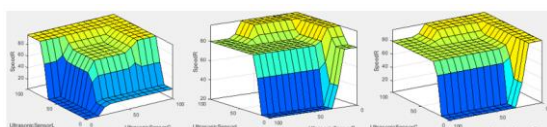


Figura 6: Relación fuzzy entre cada sensor y el motor derecho

En la Figura 7 se muestra lo mismo para el motor izquierdo. Dadas las reglas enunciadas en la Tabla 2, se obtienen relaciones diferentes para cada motor, al haber prevalecido el giro a la derecha frente al giro a la izquierda en caso de que ambas opciones fuesen igualmente válidas.

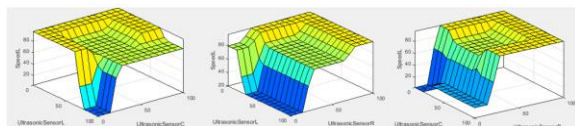


Figura 7: Relación fuzzy entre cada sensor y el motor izquierdo

Este controlador no se ha podido implementar en el microcontrolador debido a problema de memoria [2]. Las 27 reglas generan un modelo en Simulink incompatible con la memoria disponible en el microcontrolador seleccionado.

### 3.1 SEGUNDA PROPUESTA DE CONTROLADOR BORROSO

Se ha simplificado el controlador, reduciendo el número de conjuntos borrosos asociados a las entradas de los sensores, por lo que el número de reglas es menor y se adecua a la memoria del controlador. De esta forma se puede probar en el sistema real.

En este segundo caso se han definido dos conjuntos borrosos para las variables de entrada: cerca y lejos (Figura 8). La salida, que corresponde a los motores del robot móvil, no se modifica. Para realizar esta simplificación se ha tenido en cuenta el listado de

reglas descrito en la tabla 2, considerando que solo se tienen dos conjuntos borrosos y aplicando alguna de las consideraciones expresadas en [11].

El sistema de inferencia de tipo Mamdani viene definido por las ocho reglas de la Tabla 3. En este caso, según el controlador propuesto, el robot realizará la corrección de la ruta cuando esté cerca del obstáculo, siendo menos previsor que el primer controlador planteado.

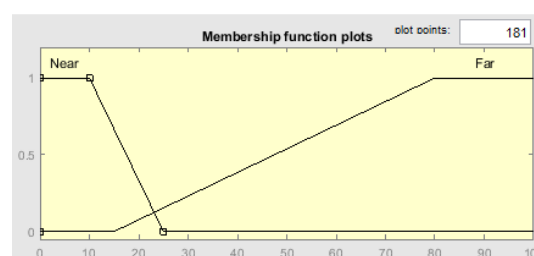


Figura 8: Conjuntos borrosos de las variables de entradas (distancia)

Tabla 3: Base de reglas del segundo controlador.

Regla	Sensor Izq	Sensor Cen	Sensor Dch	Motor Izq	Motor Dch
1	Cerca	Cerca	Cerca	Retro Rápido	Retro Rápido
2	Cerca	Cerca	Lejos	Avance Rápido	Retro Rápido
3	Cerca	Lejos	Cerca	Avance Lento	Avance Lento
4	Cerca	Lejos	Lejos	Avance Rápido	Retro Lento
5	Lejos	Cerca	Cerca	Retro Rápido	Avance Rápido
6	Lejos	Cerca	Lejos	Avance Rápido	Retro Rápido
7	Lejos	Lejos	Cerca	Retro Lento	Avance Rápido
8	Lejos	Lejos	Lejos	Avance Rápido	Avance Rápido

Las superficies de control borroso generadas para cada sensor se muestran en la Figura 9 para el motor derecho y en la Figura 10 para el motor izquierdo. Se observan relaciones similares a las obtenidas con el controlador anterior, pero con menos matices de variación. Destaca la mayor no linealidad del motor izquierdo, dadas las nuevas reglas propuestas.

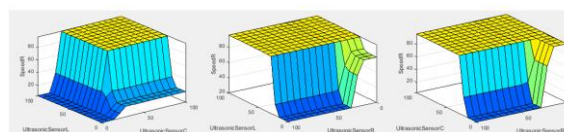


Figura 9: Relación entre cada sensor y el motor derecho

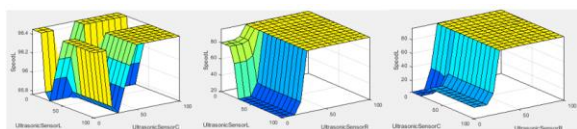


Figura 10: Relación entre cada sensor y el motor izquierdo

#### 4 RESULTADOS DE LA APLICACIÓN DE LOS CONTROLADORES BORROSOS

El comportamiento de robot se ha programado mediante la herramienta Simulink de MATLAB, que junto con la toolbox adecuada permite trabajar con Arduino. En la Figura 13 se muestra el modelo implementado en Simulink.

Los tres bloques de la izquierda reciben la información de los sensores de ultrasonidos; el bloque del controlador borroso es el diseñado en la sección anterior. Las señales PWM para controlar ambos motores, se generan a partir de una máquina de estados, la cual se ubica en ambos bloques, de la toolbox Stateflow, en color amarillo (Figura 13).

La prueba consistió en analizar el comportamiento del robot dentro de un recinto, con diferentes obstáculos. El objetivo es cruzar el mapa de la Figura 11 evitando cualquier obstáculo. Aplicando una descomposición vertical al entorno de pruebas, las trayectorias a seguir por el robot serían las descritas en esa Figura 11, lo que da lugar al grafo de trayectorias representado en la Figura 12. Cualquiera de ellas sería válida.

Para analizar los datos del entorno en tiempo real a la vez que el sistema se encuentra en funcionamiento es necesario que el robot se encuentre conectado al modelo de Simulink. Este método presenta el problema de que el sistema funciona con los pasos prefijados por el solver del modelo, y no a la frecuencia de trabajo del microcontrolador. Por ello los datos que se muestran a continuación se han

tomado con los motores desconectados. Es decir, el robot ha sido guiado en el entorno de pruebas a partir de los datos obtenidos, con el fin de verificar el comportamiento en un entorno real.

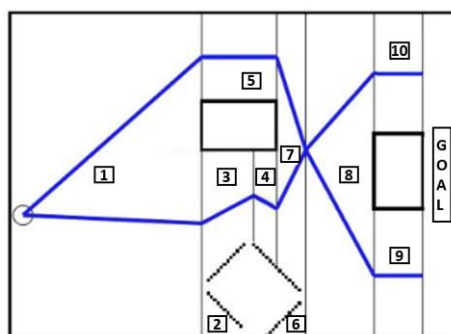


Figura 11: Representación del entorno y de la trayectoria usando descomposición vertical

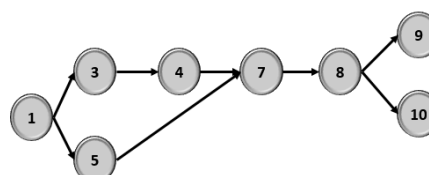


Figura 12: Grafo de trayectorias

En la Tabla 4 se muestran los datos de la lectura de cada sensor, en los puntos de interés, que permiten caracterizar la salida del controlador, es decir, el correspondiente ciclo de trabajo que determina velocidad y sentido del motor.

Para la captura de los datos, de cada sensor de ultrasonidos, recogidos en la Tabla 4, se ha implementado el modelo representado en la Figura 13, añadiéndole una serie de displays que permitan visualizar la información, en el robot propuesto en la Figura 3. No obstante, se han desconectado los motores, con el único fin de comprobar la información relevante de los sensores en las coordenadas más representativas del entorno de pruebas.

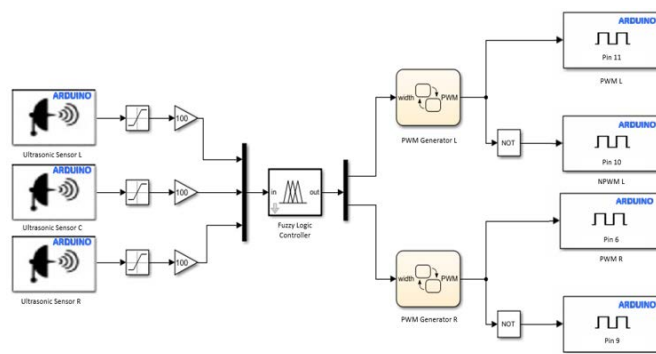


Figura 13: Modelo en Simulink del sistema de detección de obstáculos basado en lógica borrosa



Tabla 4: Datos de los sensores en puntos de interés.

Dato	Posición (x,y)	Sensor Izq	Sensor Cen	Sensor Dch
1	(62,5)	100 cm	75 cm	65 cm
2	(62,50)	100 cm	54 cm	12 cm
3	(62,65)	100 cm	100 cm	41 cm
4	(58,80)	6 cm	82 cm	51 cm
5	(50,96)	100 cm	42 cm	82 cm
6	(37,107)	79 cm	67 cm	52 cm
7	(62,96)	14 cm	68 cm	46 cm
8	(87,103)	58 cm	61 cm	54 cm
9	(95,111)	30 cm	45 cm	93 cm
10	(58,107)	23 cm	14 cm	66 cm

Estas pruebas preliminares también sirvieron para diseñar las funciones de pertenencia de las variables del sistema de control.

En nuestro caso no estamos interesados en la orientación con la que el robot móvil alcanza el objetivo final [7].

En los siguientes apartados se comprueba el comportamiento para ambos controladores propuestos. Hay que señalar que los porcentajes que aparecen tanto en la Tabla 5, así como en la Tabla 6, corresponden con el ciclo de trabajo que determinará la velocidad y sentido del robot. Un valor en el rango 0-30 % implicaría que el motor girase marcha atrás, y dentro del rango 70-100 %, que el motor girase marcha adelante. Los valores 0 % y 100 %, representan la velocidad máxima para cada caso.

#### 4.1 RESULTADOS OBTENIDOS PARA EL PRIMER CONTROLADOR

En la Tabla 5 se recoge la evolución del robot utilizando el controlador borroso de mayor complejidad, con las 27 reglas que regulan su comportamiento, para los puntos de interés del entorno de la Tabla 4.

Tabla 5: Información del primer controlador.

Dato	Posición (x,y)	Motor Izq	Motor Dch
1	(62,5)	80.0 %	96.7 %
2	(62,50)	20.0 %	96.9 %
3	(62,65)	80.0 %	97.0 %
4	(58,80)	97.0 %	20.0 %
5	(50,96)	97.0 %	80.0 %
6	(37,107)	80.0 %	96.4 %
7	(62,96)	96.3 %	20.0 %
8	(87,103)	80.0 %	80.0 %
9	(95,111)	97.0 %	80.0 %
10	(58,107)	96.4 %	3.56 %

En la Figura 14 se representa el escenario de pruebas. El origen de coordenadas se encuentra en la esquina superior izquierda.

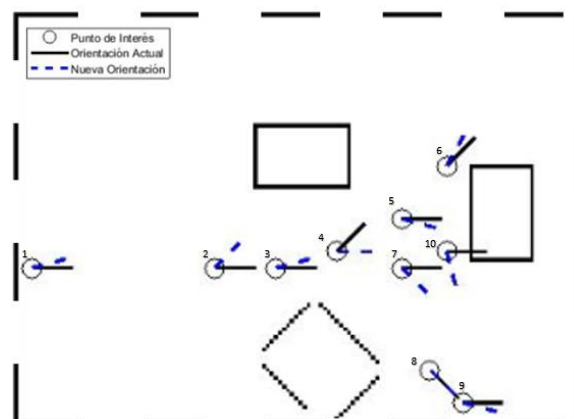


Figura 14: Representación de las orientaciones del robot en el entorno de pruebas (primer controlador)

Los puntos elegidos representan coordenadas representativas, que no se corresponden con la misma trayectoria, es decir, se evalúa el comportamiento del robot en un punto determinado. Sin embargo, a partir de este análisis puntual, es posible obtener una representación, de lo que sucedería si el robot estuviese en movimiento.

Al analizar la imagen se puede ver como en los puntos 2 y 3 reacciona al obstáculo de la derecha, pero no de la forma esperada. Según el controlador planteado, debería dar un giro moderado al principio, siendo más agresivo en caso de estar más cerca del objeto.

En realidad, el controlador está bien definido, pero, como se observa en la Tabla 3, en el punto 2 el sensor derecho registra que hay un obstáculo cerca, y en el punto 3, a media distancia. Esto manifiesta un problema en la ubicación de los sensores, pero no tiene relación con el funcionamiento del controlador borroso.

Otro punto de interés es el 6 donde, dada la colocación de los sensores, el obstáculo no se detecta y el robot sigue su curso. Es cierto que realiza una pequeña corrección, pero es debido a la naturaleza del controlador que busca la mejor trayectoria sin impedimentos. Sin embargo, lo que cabría esperar en esta situación es un giro más agresivo, que permitiese alejar al robot lo máximo posible del obstáculo.

En los puntos 8 y 9 la situación es similar; el robot mantiene una dirección que le llevará a chocarse con la pared del recinto. No obstante, en este caso el error se debe a que la barrera no era continua y presentaba huecos.

En el punto 10, se demuestra, que cuando el robot se encuentra con un obstáculo demasiado próximo realiza un giro agresivo, para ello, el motor derecho gira hacia atrás (3.56 %), y el izquierdo hacia adelante (96.4 %), a altas velocidades, permitiendo sortear el objeto.

Finalmente, para el resto de las coordenadas analizadas, el funcionamiento es el esperado y permite definir una trayectoria segura a través de los obstáculos, realizando las correcciones pertinentes mucho antes de llegar a ellos.

#### 4.2 RESULTADOS OBTENIDOS PARA EL SEGUNDO CONTROLADOR

En la Tabla 6, se recoge la evolución del robot utilizando el controlador borroso simplificado, para las mismas condiciones y consideraciones tenidas en cuenta en el caso anterior, en el entorno de pruebas propuesto.

Tabla 6: Información del segundo controlador.

Dato	Posición (x,y)	Motor Izq	Motor Dch
1	(62,5)	96.8 %	96.8 %
2	(62,50)	20.0 %	96.6 %
3	(62,65)	96.2 %	96.2 %
4	(58,80)	96.5 %	20.0 %
5	(50,96)	96.2 %	96.2 %
6	(37,107)	96.5 %	96.5 %
7	(62,96)	96.3 %	20.0 %
8	(87,103)	95.8 %	95.8 %
9	(95,111)	95.8 %	95.8 %
10	(58,107)	95.6 %	4.38 %

En la Figura 15 se muestra la respuesta del robot en cada punto de interés, con el segundo controlador, para el mismo entorno.

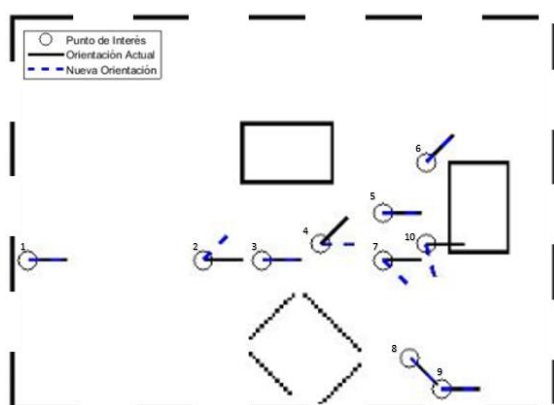


Figura 15: Representación de las orientaciones del robot en el entorno de pruebas (segundo controlador)

En este caso el comportamiento del robot es similar al anterior, con la salvedad de que con este controlador

el robot únicamente actuará cuando esté cerca del obstáculo, lo que podría comprometer la trayectoria del robot en ciertas situaciones.

## 5 CONCLUSIONES

En este trabajo se han diseñado dos controladores borrosos de distinta complejidad que son capaces de guiar a un robot móvil autónomo en su navegación en un entorno cerrado y evitar obstáculos.

El robot móvil diferencial, con ambos controladores borrosos, es capaz de seguir una de las trayectorias que le permiten alcanzar el objetivo final evitando los obstáculos del entorno.

De los dos controladores propuestos, uno es capaz de ajustar su orientación antes que el otro, permitiendo suavizar la trayectoria realizada por el robot, evitando los cambios bruscos en el momento que detecta cualquier obstáculo.

En cuanto al hardware empleado, la placa Arduino Uno ha permitido implementar un controlador sencillo, pero ha limitado la posibilidad de realizar un controlador más complejo por falta de recursos de memoria lo que ha limitado la realización de pruebas.

Para posibles trabajos futuros, sería conveniente analizar las posibilidades que ofrece la placa Arduino MEGA 2560, basada en el microcontrolador ATmega2560, que consta de cincuenta y cuatro pines E/S digitales y dieciséis pines E/S analógicos, con una memoria flash de 256 kB y una memoria EEPROM de 4 kB, con una frecuencia de 16 MHz.

MATLAB y Simulink, y las herramientas que lo componen, en concreto la Fuzzy Logic Toolbox, ha facilitado el diseño de varios tipos de controladores borrosos, ajustando los parámetros más representativos.

Como trabajos futuros se propone implementar el sistema con el hardware adecuado para la comprobación de los diferentes controladores propuestos, desarrollando un sistema de adquisición datos, que permita comprobar el sistema en las condiciones que más se asemejen a un entorno real, permitiendo realizar un mayor número de pruebas, y en diferentes condiciones. Se evaluará, no solo que el robot llegue a la meta, sino además que lo realice con la orientación adecuada.

### Agradecimientos

Este trabajo ha sido parcialmente financiado por el Proyecto RTI2018-094902-B-C21, del Ministerio de Ciencia, Innovación y Universidades.

## English summary

### DESIGN AND IMPLEMENTATION OF A FUZZY OBSTACLE DETECTION SYSTEM FOR AN AUTONOMOUS MOBILE ROBOTS

#### Abstract

*In this work we propose the design and implementation of an obstacle detection system for a mobile robot. The control system uses the information provided by ultrasonic sensors for navigation. Two controllers based on fuzzy logic have been designed, with different levels of complexity. They have been implemented in an Arduino development platform, with the software MATLAB/Simulink, and it has been tested in a real system. The results of the experiments with the robot have proven that the fuzzy control allows the robot to follow a trajectory avoiding obstacles even in relatively complex scenarios.*

**Keywords:** Intelligent control, fuzzy logic, mobile robot, obstacle detection, navigation.

#### Referencias

- [1] Berisha, J., Bajrami, X., Shala, A., Likaj, R. (2016, June). Application of Fuzzy Logic Controller for obstacle detection and avoidance on real autonomous mobile robot. In *2016 5th Mediterranean Conference on Embedded Computing (MECO)*(pp. 200-205). IEEE.
- [2] Bravo, F. G., García, J. M., Naharro, R. J., Galán, J. G., Raya, M. S. (2017). Plataforma experimental para el estudio de la vulnerabilidad hardware en los robots móviles: el Bus I2C como caso de estudio. *Revista Iberoamericana de Automática e Informática industrial*, 14(2), 205-216.
- [3] Guzmán, L. E. S., Villa, M. A. M., & Vásquez, E. L. R. (2014). Seguimiento de trayectorias con un robot móvil de configuración diferencial. *Ingenierías USBMed*, 5(1), 26-34.
- [4] Jantzen, J. (2007). *Foundations of fuzzy control* (Vol. 209). West Sussex: John Wiley & Sons.
- [5] Lafta, H. A., & Hassan, Z. F. (2015). Mobile robot control using fuzzy logic. *Journal of University of Babylon*, 23(2), 524-532.
- [6] Marichal, G. N., Acosta, L., Moreno, L., Méndez, J. A., Rodrigo, J. J., & Sigut, M. (2001). Obstacle avoidance for a mobile robot: A neuro-fuzzy approach. *Fuzzy Sets and Systems*, 124(2), 171-179.
- [7] Masmoudi, M. S., Krichen, N., Masmoudi, M., Derbel, N. (2016). Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Applied Soft Computing*, 49, 901-919.
- [8] Peri, V. M., & Simon, D. (2005, June). Fuzzy logic control for an autonomous robot. In *Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American* (pp. 337-342). IEEE.
- [9] Omrane, H., Masmoudi, M. S., Masmoudi, M. (2016). Fuzzy logic based control for autonomous mobile robot navigation. *Computational Intelligence and Neuroscience*, 2016.
- [10] Santos, M. (2011). Un enfoque aplicado del control inteligente. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(4), 283-296.
- [11] Yerubandi, V., Reddy, Y. M., & Reddy, M. K. (2015). Navigation system for an autonomous robot using fuzzy logic. *International journal of scientific and research publications*, 5(2).



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).