

Sistema de navegación y evasión de obstáculos aplicando un sistema de control difuso en una placa Arduino UNO

David Jesús Segura Cristino, Herón Molina Lozano, Elsa Rubio Espino,
Víctor H. Ponce Ponce

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México
absol.1994@gmail.com, hmolina@cic.ipn.mx, erubio@cic.ipn.mx

Resumen. La navegación autónoma de robots móviles se ha convertido en un área de gran interés, debido a su presencia en diversas áreas industriales. En este trabajo se propone la implementación de un sistema de control difuso tipo Mamdani que utiliza la defuzzificación del Método del Centro de Máximos (COM), en una plataforma con recursos computacionales limitados (arduino UNO); demostrando así, que es posible implementar controles difusos en sistemas de poco poder computacional sin tener que implementar un sistema de comunicación a una plataforma con mayores recursos (computadora).

Palabras clave: conjuntos difusos, lógica difusa, control difuso, robot diferencial, navegación autónoma.

Navigation and Obstacle Avoidance System Applying a Fuzzy Control System on an Arduino UNO

Abstract. Autonomous navigation of mobile robots has become an area of great interest, due to its presence in various industrial areas. This paper proposes the implementation of a Mamdani type fuzzy control system that uses the defuzzification of the Maximum Center Method (COM), on a platform with limited computational resources (arduino UNO); thus demonstrating that it is possible to implement fuzzy controls in systems of little computing power without having to implement a communication system to a platform with greater resources (computer).

Keywords: fuzzy sets, fuzzy logic, fuzzy control, differential robot, autonomous navigation.

1. Introducción

Actualmente la mayoría de las industrias están empleando robots para la realización de tareas, ya que estos poseen precisión y repetibilidad en sus acciones; razón por la cual es de gran importancia el desarrollo constante de nuevos

algoritmos de control que se adapten a los constantes cambios de tecnología y que se cumplan con los requerimientos eficiencia que se necesitan en cualquier tipo de aplicaciones industriales, de medicina, etc. La capacidad de un robot móvil de posicionarse de manera correcta es muy importante para que éste sea capaz de alcanzar un objetivo o que pueda evadir obstáculos de un ambiente industrial o de difícil acceso [1].

Por otro lado, es ya conocido que el control difuso nos permite implementar un control con razonamiento similar al de un ser humano, en un sistema computacional. Es muy útil cuando el sistema a controlar es complejo y no hay un modelo matemático o la implementación de dicho modelo consume muchos recursos, ya que el control difuso no requiere de ningún modelo para su uso. Este tipo de sistemas se pueden mejorar fácilmente pues el cambio de algún conjunto de entrada no afecta a las normas de control o al resto de los conjuntos de entrada/salida [1].

Por lo dicho anteriormente, en este trabajo se propone un sistema de control para la navegación y evasión de obstáculos de un robot diferencial no holonómico (RDNH) utilizando un sistema de control difuso tipo Mamadani.

Este artículo está organizado de la siguiente manera: el estado del arte se presenta en la sección II, el desarrollo teórico en la sección III, el desarrollo experimental se encuentra en la sección IV, seguido de la discusión de resultados en la sección V y finalmente se presentan las conclusiones.

2. Estado del arte

En el pasado se han realizado otros trabajos para comprobar el desempeño de diversos métodos para el control de robots móviles. Por ejemplo, en [3] se propone un sistema de control basado en la función de Lyapunov y para su implementación se tienen que encontrar tanto las variables a controlar, como las ecuaciones de la velocidad lineal y angular del robot, las ecuaciones de las velocidades angulares de los motores derecho e izquierdo y finalmente se usa odometría para conocer la posición del robot. Si se compara lo anterior con un sistema difuso, puede apreciarse que la dificultad de implementación en un sistema difuso es mucho menor.

El trabajo realizado en [1] trata de un sistema de control de navegación parecido al propuesto en este artículo, en el que se usa la lógica difusa y en donde se observa que la implementación del sistema es bastante sencilla, obteniendo buenos resultados en la simulación numérica del sistema. Un sistema de evasión de obstáculos que hace uso de la lógica difusa es [6], el artículo únicamente trata el problema de la evasión de obstáculos, por lo anterior no se implementa ninguna forma de navegación para llegar a un objetivo. En [4] se propone un sistema en chip que usa lógica difusa para navegación, este sistema está montado en un FPGA, por lo que aumenta la velocidad de procesamiento del sistema de control. El trabajo en [7], usa un sistema neuro-difuso para la evasión de obstáculos y esta implementado en una computadora la cual se conecta inalámbricamente con un robot que también le comunica las lecturas de sus

sensores. Una implementación similar a la propuesta en este trabajo, es la presentada en [2]; en donde los comportamientos de evasión y navegación utilizan un control difuso. La implementación del sistema se realizó en un robot Scout-II, que a su vez se conecta inalámbricamente a un *host* WiFi, en él se encuentra montado un servidor con el sistema de control, lo anterior es posible debido a que la velocidad de comunicación es suficiente para que sistema responda oportunamente ante cualquier situación que se presente en el recorrido.

En los trabajos anteriores, la mayoría de los sistemas requieren de un sistema externo para montar el control difuso que se comunica con el robot, aprovechando así la facilidad de programación y los recursos computacionales de los que podría carecer el robot móvil que se controla. A pesar de lo anterior, hay ejemplos en [6], [8], [5], en los cuales se han implementado controles difusos sencillos en placas Arduino las cuales carecen de los recursos de una computadora; pero que se podrían conectar al sistema a controlar mediante WiFi. Tomando como base lo anteriormente dicho, en este trabajo, se usará también una placa Arduino para la implementación del sistema de control difuso propuesto. Ver sección 4.

3. Desarrollo teórico

3.1. Odometría

Ya que se usará control difuso, no se requiere un modelo del sistema, sin embargo, el vehículo requiere conocer su ubicación en todo momento para saber cómo debe comportarse y cómo afectan las decisiones que tome el sistema de control. La odometría permite conocer la posición del robot móvil. Se usará *dead-reckoning* que es un método con base en la posición y orientación previas al tiempo $t = t_k$. Usando una aproximación de Euler en el modelo cinemático del robot diferencia (ver ref.[2]).

$$\begin{cases} X(k+1) = X(k) + v(k)T_s \cos(\theta(k)), \\ Y(k+1) = Y(k) + v(k)T_s \sin(\theta(k)), \\ \theta(k+1) = \theta(k) + \omega(k)T_s. \end{cases} \quad (1)$$

donde $T_s = T_{k+1} - T_k$ es el tiempo de muestreo. Lo más conveniente es usar la distancia que se desplazó el robot durante el tiempo de muestreo, esto puede ser calculado usando un *encoder* para conocer el desplazamiento angular de cada llanta, otro método consiste en inferir dicho desplazamiento usando la velocidad de desplazamiento a la que gira el motor, para esto se requiere que la velocidad a la que gira el motor se pueda controlar, cómo se hace en un servomotor o un motor paso a paso. Por lo anterior se modificó la ecuación 1, quedando como sigue:

$$\begin{cases} X(k+1) = X(k) + \frac{\pi D}{2} (\Delta T_l + \Delta T_r) \cos(\theta(k)), \\ Y(k+1) = Y(k) + \frac{\pi D}{2} (\Delta T_l + \Delta T_r) \sin(\theta(k)), \\ \theta(k+1) = \theta(k) + \frac{\pi D}{2} (\Delta T_r - \Delta T_l). \end{cases} \quad (2)$$

donde $\Delta T = T_{k+1} - T_k$, es la cantidad de revoluciones que realiza una rueda en el tiempo de muestreo (l se refiere a la rueda izquierda y r , se refiere a la rueda derecha).

A grandes velocidades el *Dead Reckoning*, muestra poca precisión en la estimación de posiciones pero a velocidades bajas demuestra un buen desempeño [2]. Este es el método que se uso en el desarrollo experimental de este trabajo para conocer la posición del robot.

3.2. Comportamientos

Para que el robot sea capaz de navegar hasta un obstáculo y además pueda esquivar obstáculos, se requiere que tenga ambos comportamientos programados y sea capaz de alternar entre ambos, para ésto se usaron sensores ultrasónicos que permiten reconocer si hay un obstáculo frente al robot y a qué distancia se encuentra el robot de él. Cuando los sensores envían una señal de una presencia en cierto rango activarán el comportamiento que permite esquivar obstáculos del robot y el robot comenzará a moverse de acuerdo a lo dictado por dicho comportamiento y a su vez le permitirá alejarse o evitar el obstáculo. Mientras no haya nada que le impida el paso, el robot estará usando el comportamiento de navegación, que lo lleva a buscar la ruta más corta a un objetivo definido. Alternando entre ambos comportamientos el robot es capaz de desenvolverse en un entorno completamente desconocido. Una descripción de la implementación de ambos comportamientos se puede ver en la figura 1.

3.3. Navegación

El comportamiento de navegación es el que hace que el robot se mueva hacia el objetivo para que el robot cumpla con su misión. Las entradas de este sistema son dos y los vemos representados en la figura 2, la distancia desde la posición del robot hasta el objetivo, la cual se calcula de la siguiente forma:

$$P = \sqrt{(X_r - X_t)^2 + (Y_r - Y_t)^2}, \quad (3)$$

donde X_r, Y_r es el vector de posición del robot, y X_t, Y_t es el vector que describe la posición del objetivo a alcanzar del robot.

La segunda variable de entrada de este control es el error de ángulo α :

$$\alpha = \beta - \theta, \quad (4)$$

donde β es el ángulo del vector de distancia del robot al objetivo y θ es la orientación actual del robot.

La salida del sistema son las velocidades de giro de los motores w_R y w_L que son la velocidad angular de la llanta derecha y la velocidad angular de la llanta izquierda respectivamente. Los motores usados son motores Dynamixel AX-12A los cuales no reciben velocidades angulares, sino valores que van desde -1023 a 1023, por esto la variable de salida no será dada en rad/s , sino en valores con

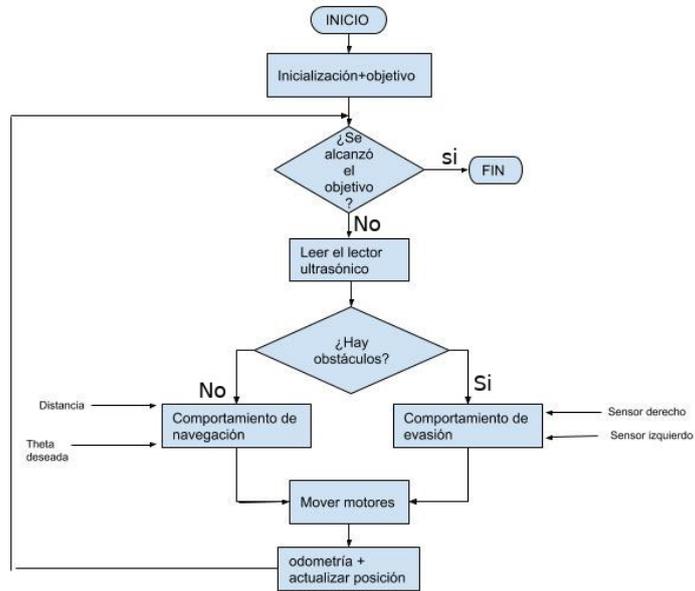


Fig. 1. Diagrama de flujo del comportamiento del robot móvil.

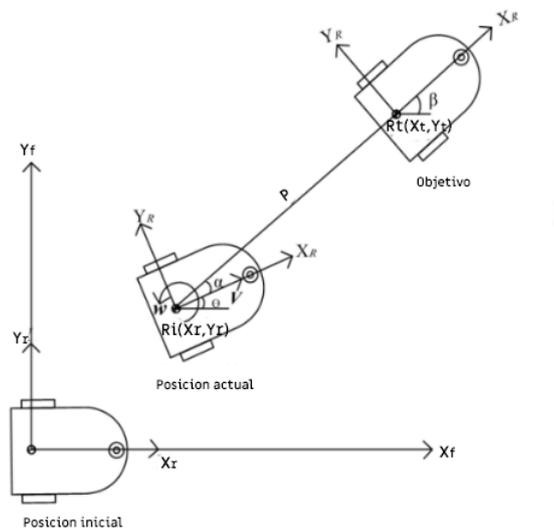


Fig. 2. Distancia al objetivo (P) y error de orientación (α) [3].

el mismo rango al que trabajan los motores. Lo anterior es una ventaja pues no es necesario calcular las salidas en velocidad angular y realizar conversiones de

rad/s al valor numérico que representaría dicha velocidad en los motores, lo cual acarrearía errores que se acumularían en cada iteración del comportamiento.

Se usaron cinco conjuntos difusos para las variables de entrada, esto nos permite tener un mejor control y precisión. Para los conjuntos de la variable P se usaron cuatro conjuntos triangulares para las distancias cercanas al robot móvil y un trapezoidal para el conjunto que representa la distancia más lejana. En cuanto a la variable de control α se usan cinco conjuntos triangulares y su rango de operación va de $-\pi$ a π . En las figuras 3 y 4 se pueden ver las gráficas que muestran éstos conjuntos.

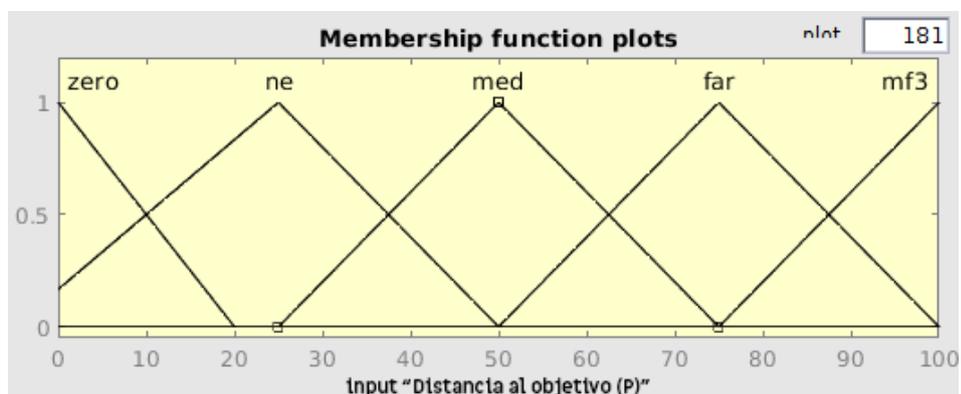


Fig. 3. Distancia al objetivo (P).

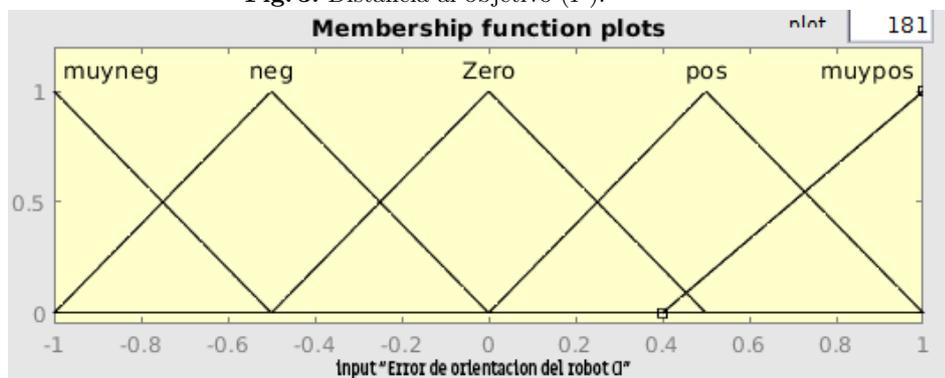


Fig. 4. Error de orientación del robot α .

Como ya se mencionó, las variables de salida no serán dadas en rad/s, sino en valores entre -1023 y 1023. La velocidad a la que se mueven los motores cuando se les da un valor de 1023 es muy alta y podría generar errores, por lo anterior el rango de operación de los motores se reducirá a la mitad. En las figuras 5 y 6 podemos ver cómo son dichos conjuntos.

Ya que tenemos cinco conjuntos de cada variable de entrada, se generan veinticinco reglas difusas *si-entonces* las cuales se muestran en la figura 7.

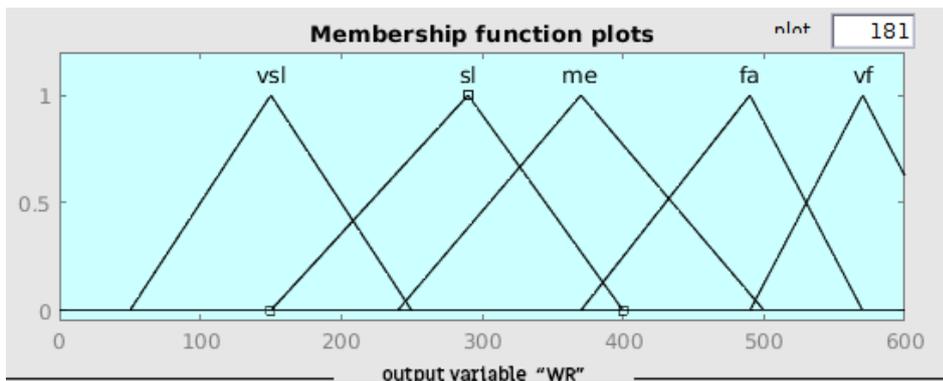


Fig. 5. Velocidad de movimiento del motor Derecho (wr).

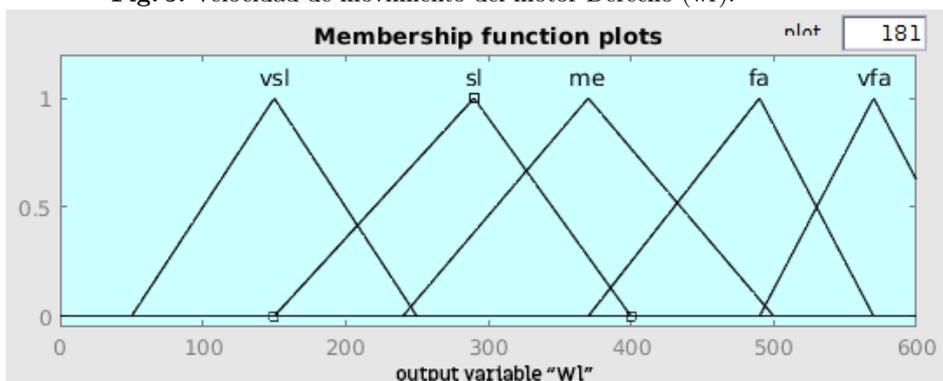


Fig. 6. Velocidad de movimiento del motor Izquierdo (wl).

3.4. Evasión de obstáculos

La evasión de obstáculos por parte del robot, ocurre cuando los sensores ultrasónicos detectan que hay un obstáculo que impide su avance, y que además se encuentra dentro de la ruta generada por el comportamiento de navegación. Sus variables de entrada son las distancias medidas por los sensores ultrasónicos (D_l y D_r). Las distancias están expresadas en *cm* para facilidad del usuario, y sus variables de salida son las velocidades de los motores, los cuales, al igual que en comportamiento de navegación se encuentran entre -600 y 600 para evitar que el robot se mueva velocidades muy altas generando errores en la odometría.

Puesto que en este caso ambas variables tienen el mismo rango de trabajo, pero en posiciones diferentes, se pueden usar los mismos conjuntos difusos para

alphaP	zero	ne	med	far	veryfar
muyneg	Rvsl	Rvsl	Rsl	Rm	Rm
	Lm	Lm	Lf	Lf	Lvf
neg	Rvsl	Rvsl	Rvs	Rs	Rsl
	Lsl	Lsl	Lm	Lf	Lf
Zero	Rvsl	Rsl	Rm	Rf	Rvf
	Lvsl	Lsl	Lm	Lf	Lvf
pos	Rsl	Rsl	Rm	Rf	Rf
	Lvsl	Lvsl	Lvsl	Lsl	Lsl
muypos	Rm	Rm	Rf	Rf	Rvf
	Lvsl	Lvsl	Lsl	Lm	Lsl

Fig. 7. Reglas difusas de navegación.

ambas. Estos son cinco conjuntos trapezoidales que van desde 0 *cm* hasta 200 *cm* y se muestra en las figuras 8 y 9; aunque para mostrarlas correctamente sólo se muestra el rango de 0 a 50 *cm*.

Los conjuntos difusos de las variables de salida se pueden ver en las figuras 10 y 11.

En este caso se generarán nueve reglas difusas que definirán cómo responde este sistema de control figura 12.

3.5. Robot móvil

Para realizar las pruebas con el sistema de control se usó un robot construido con el kit Bioloid (figura 13), disponible en el laboratorio de Robótica y Mecatrónica del CIC, se reemplazó el controlador que venía con el kit Bioloid y en su lugar se usó una tarjeta de desarrollo Arduino UNO.

Los motores empleados para el movimiento del robot son dos motores AX-12A, los cuales se comunican al Arduino mediante comunicación serial, que funcionan de 7v a 12v. Para controlar la velocidad se les debe indicar un número en un rango de -1023 a 1023, mismo que representa los valores de la velocidad máxima de giro del motor.

Para detectar la presencia de obstáculos y la distancia a la que se encuentran estos del robot, se usó el sensor para Arduino HC-SR04.

3.6. Defuzificación

El método de defuzificación más empleado es el método del centroide, pues permite un control más preciso, sin embargo este método implica el cálculo del área de los conjuntos difusos y esto va más allá de las capacidades del Arduino UNO, es por esto que para hacer la defuzificación se usó el método centro de

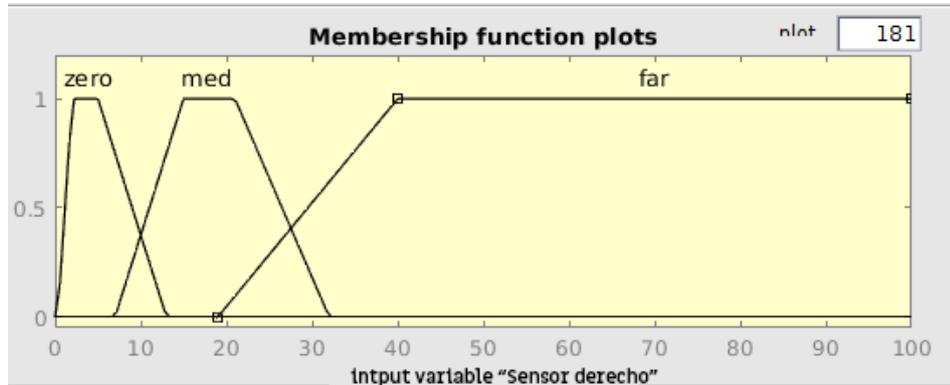


Fig. 8. Distancia al obstáculo sensor derecho.

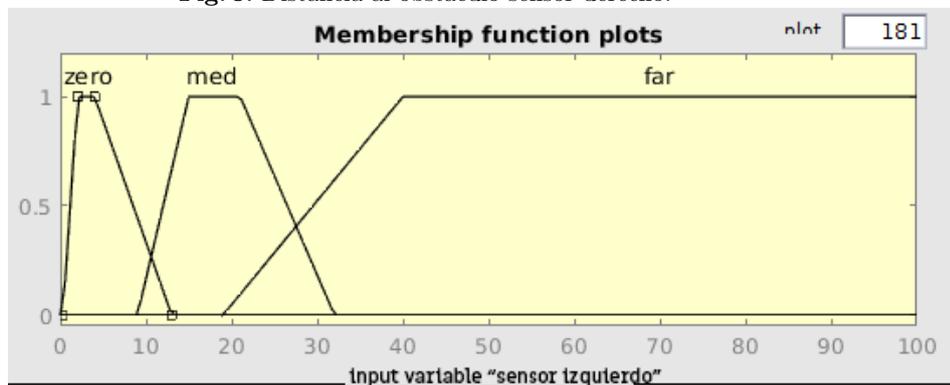


Fig. 9. Distancia al obstáculo sensor izquierdo.

máximos (COM), que se usó en [8], pues es un método que requiere pocos recursos y operaciones algebraicas que fueron programadas en el Arduino UNO.

$$Z^* = \frac{\sum_{i=1}^n \mu(ui) * ui}{\sum_{i=1}^n \mu(ui)}, \quad (5)$$

donde ui es el valor típico de cada set difuso y $\mu(ui)$ es el valor de pertenencia en ese conjunto difuso.

El método COM es eficiente, si se utilizan conjuntos difusos simétricos, además permite obtener salidas reales usando recursos computacionales limitados [8].

4. Desarrollo experimental

Se anexa un link a un video del robot en funcionamiento:

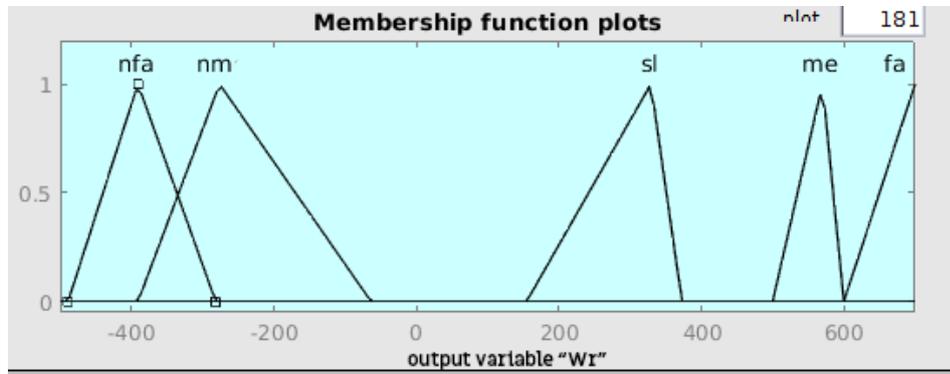


Fig. 10. Velocidad de movimiento del motor Derecho (wr).

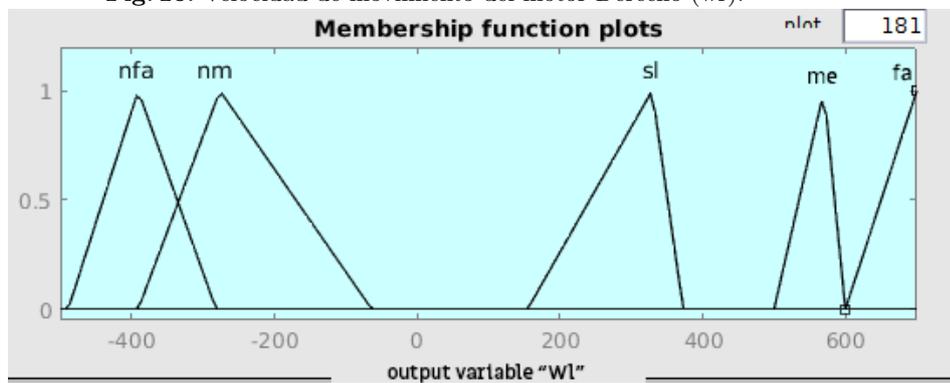


Fig. 11. Velocidad de movimiento del motor Izquierdo (wl).

Dr/Dl	zero	med	far
zero	Rnf	Rsl	Rns
	Lnf	Lnf	Lnf
med	Rnf	Rf	Rme
	Lsl	Lme	Lf
far	Rnf	Rme	Rf
	Lns	Lf	Lf

Fig. 12. Reglas difusas de evasión.

<https://drive.google.com/file/d/1oTFy7lFWEoDNotDydwssSqNKfAXcAcFT/view?usp=sharing>

Se realizaron algunas pruebas con el sistema de control implementado en el robot diferencial con el Arduino UNO (figura 14). En las pruebas el robot

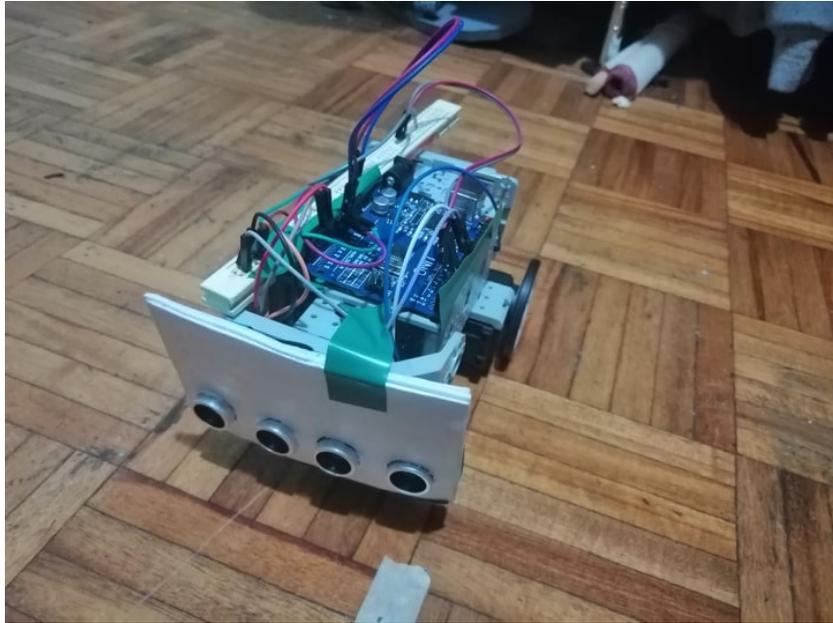


Fig. 13. Robot usado en las pruebas.

fue capaz de llegar a la posición objetivo sin colisiones con los obstáculos y en un tiempo corto (menos de un minuto en las pruebas realizadas). El área en el que se realizaron las pruebas fue en un escritorio del laboratorio de robótica y mecatrónica del CIC, esto debido a las limitaciones de la alimentación del sistema, se usó una fuente de 9v de CD conectada a la toma de corriente.

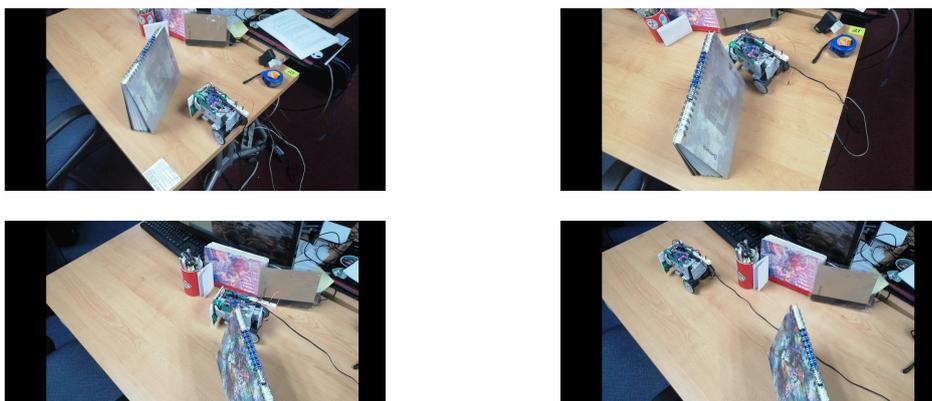


Fig. 14. Imágenes de las pruebas realizadas.

En dos pruebas el robot colisionó con los obstáculos (figura 15(a)) que se encontraban en el área de pruebas. Debido a la ubicación de los sensores ultrasónicos, el robot tiene problemas con los obstáculos que se encuentran fuera del rango de visión de los sensores, pero que están lo suficientemente cerca como para chocar con sus llantas (figura 15(b)), que por el diseño del robot, se encuentran más alejadas del centro de lo que están los sensores.



a) Choque del vehículo con un obstáculo.



b) El Robot móvil tiene problemas con obstáculos muy pegados a sus costados.

Fig. 15. Problemas encontrados durante las pruebas del Robot diferencial.

5. Discusión

Los resultados experimentales demuestran que es posible implementar un sistema de control difuso tipo Mamdani en una tarjeta embebida como el Arduino UNO. Manteniendo los beneficios del uso del control difuso sin usar grandes recursos de *hardware* o de realizar los cálculos del control en una computadora. Todo lo anterior, se reduce sólo a enviarle las variables de velocidad al robot.

Si bien en algunas ocasiones el sistema falló (ver figura 2) y el robot chocó con los obstáculos, ésto se puede llegar a solucionar modificando los conjuntos propuestos o las reglas, lo cual no representa ningún problema, pues la facilidad de mejorar el sistema es una de las ventajas de los sistemas de control difusos.

6. Conclusiones

Si bien los sistemas de control difusos tipo Mamdani pueden llegar a requerir muchos recursos computacionales, es posible y es sencillo el implementarlos en sistemas de bajo poder computacional como un Arduino UNO, si bien se pierde precisión en los cálculos comparando los resultados con un *hardware* con más recursos, ésto se puede compensar modificando la forma de los conjuntos difusos de entrada. Trabajar el control difuso de esta forma permite que no sea necesario implementar un sistema de comunicación inalámbrica o alámbricamente con una computadora, ni cambiar la placa de control por un sistema más poderoso como una Raspberry o una Beaglebone.

Como trabajo futuro se planea refinar este sistema de control modificando los conjuntos difusos, rediseñando el robot, cambiando la ubicación de los sensores y las dimensiones de este.

Referencias

1. Rashid, R., Elamvazuthi, I., Begam, M., Arrofiq, M.: Differential Drive Wheeled Mobile Robot (WMR) Control Using Fuzzy Logic Techniques. In: 2010 Fourth Asia International Conference on Mathematical Analytical Modelling and Computer Simulation, 51–55 (2010)
2. Faisal, M., Hedjar, R., Al Sulaiman, M., Al-Mutib, K.: Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment. International Journal of Advanced Robotic Systems (2013)
3. Sandeep Kumar Malu, B., Majumdar, J.: Kinematics, Localization and Control of Differential Drive Mobile Robot. Global Journal of Researches in Engineering: H Robotics & Nano-Tech, 14(1), 9 (2014)
4. Tzafestas, S. G., Deliparaschos, K. M., Moustiris, G. P.: Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of System on a Chip. Robotics and Autonomous Systems, 58(8), 1017–1027 (2010)
5. Jayetileke, H. R., de Mei, W. R., Ratnayake, H. U. W.: Real-time fuzzy logic speed tracking controller for a DC motor using Arduino Due. In: 7th International Conference on Information and Automation for Sustainability, 1–6 (2014)
6. Bobyr, M. V., Kulabukhov, S. A., Milostnaya, N. A.: Fuzzy control system of robot angular attitude. In: 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 1–6 (2016)
7. Zerfa, H., Nouibat, W.: Fuzzy reactive navigation for autonomous mobile robot with an offline adaptive neuro fuzzy system. 3rd International Conference on Systems and Control, 950–955 (2013)
8. Espino Núñez, A., Ramírez Martínez, F.: Control de temperatura con lógica difusa para un sistema de espectroscopía láser. Tesis de la Facultad de Ingeniería, Universidad Autónoma de México (2017)