

## **Metodología para controlar un robot móvil con lógica difusa**

David Luna, Ana C. Zenteno, María del C. Santiago, Yeiny Romero,  
Judith Pérez, Gustavo T. Rubín

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
México

divanso@hotmail.com, {ana.zenteno,  
marycarmen.santiago, yeiny.romero, judith.perez,  
gustavo.rubin }@correo.buap.mx

**Resumen.** En este documento se describe el proceso para lograr el control de un robot móvil Scribbler con lógica difusa. Inicialmente se desarrolló un simulador para este proyecto y se implementaron los algoritmos en amplitud y primero el mejor. Se utilizan estos algoritmos al ser algoritmos de búsquedas sin información y con información del SLAM. Paso siguiente, se postula una poda de los nodos cercanos a los objetos del mapa utilizando una función triangular. Posteriormente se utilizan las variables independientes distancia y velocidad para crear un par de conjuntos difusos que se implementan en el modelo de Sugeno [13], se obtiene la variable dependiente frenar y finalmente se realizó el proceso del controlador difuso para desarrollar un algoritmo de búsqueda con lógica difusa.

**Palabras clave:** Controlador Difuso, Lógica Difusa, SLAM, Robot Scribbler.

## **Methodology for Control a Mobile Robot with Fuzzy Logic**

**Abstract.** This document describes the process to achieve control of a Scribbler mobile robot with fuzzy logic. Initially a simulator was developed for this project and the algorithms were implemented in amplitude and first the best. These algorithms are used as they are search algorithms without information and with information from the SLAM. Next step, a pruning of the nodes near the map objects is postulated using a triangular function. Subsequently, the independent variables distance and speed are used to create a pair of fuzzy sets that are implemented in Sugeno's model [13], the dependent variable brake is obtained and finally the fuzzy controller process was carried out to develop a search algorithm with diffuse logic.

**Keywords:** Fuzzy Controller, Fuzzy Logic, SLAM, Robot Scribbler.

## 1. Introducción

Los primeros robots móviles se desarrollaron como una estructura mecánica anclada en uno de sus extremos. La robótica es la ciencia que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano, esta rama computacional trabaja en la autonomía de los móviles. Los robots móviles tienen como precedente los dispositivos electromecánicos; tales como el denominado “micro mouse”, creado para desarrollar funciones como descubrir caminos en laberintos [7]. En los años setentas aparecieron robots móviles dotados de autonomía, basados en sistemas de visión [8]. Posteriormente se crearon para interiores por parte de Bares, realizados en el Carnegie Mellon University de Pittsburg EEUU [9]. El SLAM se presentó por primera vez en un panfleto sobre robots móviles en el simposio internacional de investigación robótica de 1995 [10]. Se alcanzó un acercamiento entre el SLAM basado en el filtro Kalman y los métodos probabilísticos para la localización y mapeo introducidos por S. Thurn, en la conferencia internacional de robótica y automatización (ICRA) del Institute of Electrical and Electronics Engineers (IEEE) de 1999. El trabajo con SLAM atrajo a 15 investigadores que buscaban soluciones para la complejidad, la asociación de la información y retos de la implementación [11]. En el año de 2012 se empezó a usar el Kinect en el SLAM, en el proyecto localización y generación de mapas del entorno de un robot utilizando este dispositivo de la Universidad Politécnica de Valencia [12]. El concepto de conjuntos difusos fue propuesto por el matemático Lofti Ali Asker Zade en la Universidad de California, Berkeley, la lógica clásica es precisa, a diferencia de la lógica difusa, esta última se basa en la percepción. El objetivo de esta técnica es imitar la toma de decisiones del cerebro humano [4].

John McCarthy en Dartmouth, acuñó por primera vez el concepto de inteligencia artificial como la ciencia de hacer máquinas inteligentes [1], la cual es la base de la tercera revolución industrial formulada por Jeremy Rifkin [2]. Por otro lado, la lógica difusa se fundamenta en determinar el razonamiento y pensamiento humano basándose en la percepción [3], con el objetivo de imitar la toma de decisiones del cerebro humano [4], este conjunto se caracteriza por la función de pertenencia de la ecuación 1:

$$\mu_A : U \rightarrow [0,1], \quad (1)$$

donde a cada elemento  $x$  de  $U$  se le asocia un número  $\mu_A(x)$  del intervalo  $[0,1]$ , que representa el grado de pertenencia de  $x$  al conjunto difuso  $A$  y  $U$  es la *universo de discurso*, que se utiliza para ser implementado en el controlador difuso compuesto por

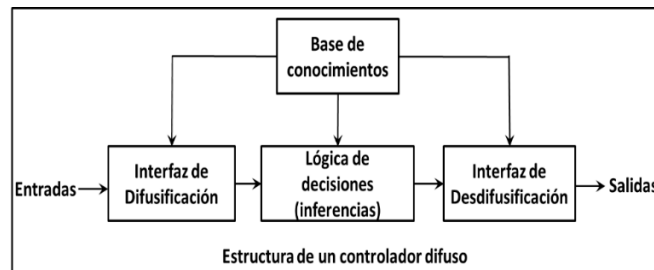


Fig. 1. Estructura de un controlador difuso.

interfaz de defusificación, base de conocimiento, lógica de decisiones e interfaz de desdifusificación (Fig. 1).

## 2. Trabajos relacionado

En 2007 se describe la navegación de robots móviles mediante comportamientos utilizando lógica difusa. Para la implementación se utilizó el toolbox de lógica difusa de Matlab y el modelo cinemático de la plataforma P-METIN del Grupo GIROPS [17]. En 2009 se presenta el diseño e implementación de un sistema de visión por computadora y control difuso orientado a la navegación autónoma de un robot móvil Lego NXT con capacidades de localización y mapeo simultáneo en un ambiente controlado mediante un algoritmo de navegación con atareas como la búsqueda de la ruta más cercana mediante el algoritmo de Dijkstra, la detección y evasión de los obstáculos presentes en su entorno y la alineación en el desplazamiento de la ruta basada en visión y lógica difusa [18]. Otro trabajo que hace referencia al navegador difuso de un robot móvil, utiliza lógica difusa para controlar el avance del robot hacia una ubicación específica con obstáculos [19]. Un trabajo más es una plataforma robótica móvil autónoma con ruedas mecanum para estudios en ambientes de carga, se analizan los posibles movimientos y se eligen los que van a ser estandarizados dentro del sistema. Se utilizan ecuaciones cinemáticas y de lógica difusa a bajo nivel para controlar la velocidad de los motores de las ruedas [10].

Otros trabajos que combina la lógica difusa con la Inteligencia Artificial son: La aplicación de técnicas de inteligencia artificial para la navegación de robots móviles utilizando el simulador player/stage [14] y la localización basada en lógica difusa y filtros de Kalman para robots con patas [15]. De la misma forma en la inteligencia artificial se han realizado trabajos de investigación utilizando los algoritmos genéticos, como se puede observar en el proyecto de Lozano Márquez Aplicación de técnicas basadas en lógica difusa para la mejora del comportamiento de los algoritmos genéticos con codificación real [16].

## 3. Metodología

Para esta investigación se estudiaron las variables físicas del Robot, como el frenado del robot, la forma de frenado, la distancia y velocidad en el paro del móvil y la forma de actuar de sus sensores, en la siguiente etapa se desarrolló un simulador con el objetivo de implementar los algoritmos en amplitud y primero el mejor, en este código se analizaron ambos algoritmos, se formuló una poda de los nodos que no se pueden explorar por el robot; posteriormente se plantea un par de variables independientes a las cuales se les dio el nombre de distancia y velocidad que se utilizaron para formular un par de conjuntos difusos, que se estudiaron para el desarrollo del controlador difuso, lo anterior fue para lograr obtener una variable dependiente que se le llamó *Frenar* que se utiliza para lograr que el móvil reduzca su velocidad, en la última etapa de este trabajo de investigación se utilizaron los resultados del control difuso para formular un

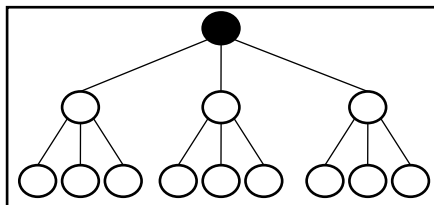


Fig. 2. Árbol con tres hijos.



Fig. 3. Interfaz de simulador.

nuevo algoritmo que ayudará en el mapeo de una región y que es un aporte para el SLAM.

#### 4. Desarrollo

Para el desarrollo de este trabajo se utilizó el modelo del frenado del robot Scribbler® obtenido en trabajos previos [7]. Posteriormente, tomando como base las características físicas del Robot, él cuenta con tres sensores de luz colocados de la siguiente forma, uno colocado en la parte izquierda, parte central y el último en la parte derecha del móvil [8], por lo cual la estructura de datos llamada árbol 3-ario representada en Fig. 2 es la adecuada para representar sus sensores de luz con los que cuenta el robot móvil. La raíz del árbol es el punto de partida donde se colocará el móvil; para el desarrollo de los algoritmos fue necesario se implementara una estructura de datos llamada pila, en donde se guardarán los nodos visitados.

El proyecto se implementa en java por ser un lenguaje de alto nivel, (se puede utilizar cualquier lenguaje para desarrollar el simulador) para desarrollar un simulador cuya interfaz gráfica se ha representado en la Fig. 3. En el simulador se implementó la búsqueda en amplitud, Fig. 4, y la búsqueda el primero el mejor, Fig. 5.

Se realizó un estudio de los algoritmos de búsqueda sin información e información, con el objetivo de mejorar sus resultados. Se puede observar en la Fig. 3 que existen nodos que se encuentran muy cerca de los objetos del mapa, los cuales al realizar un mapeo crean colisiones como se pueden estudiar en la Fig. 6. Enseguida se formula la solución de esta problemática.

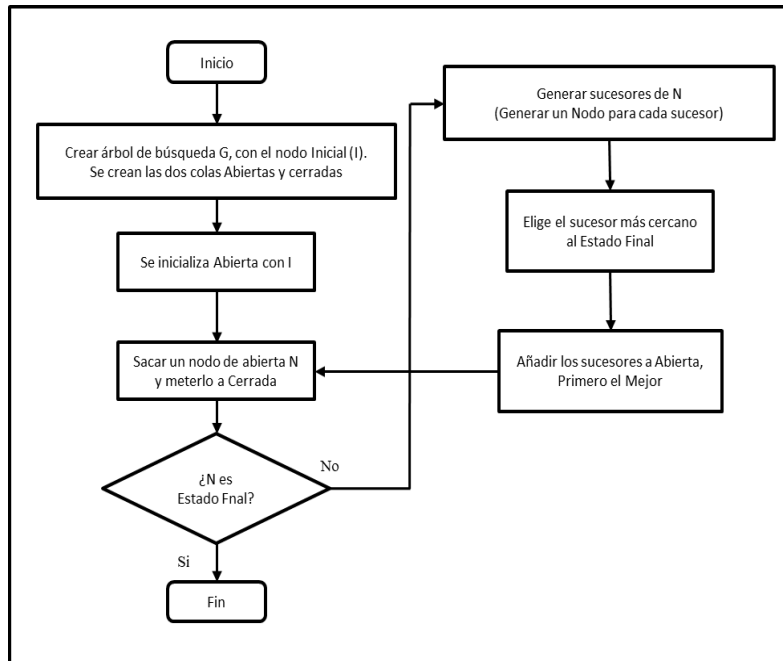


Fig. 4. Búsqueda por Amplitud.

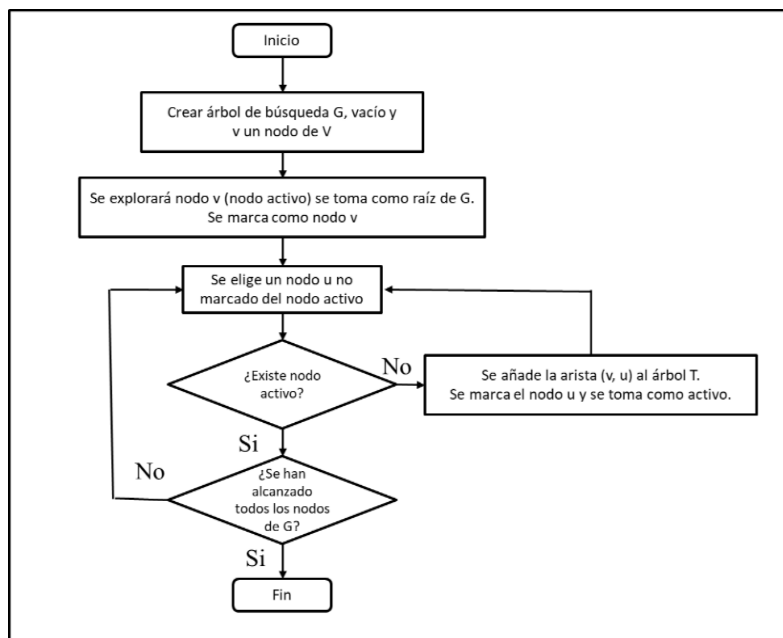


Fig. 5. Búsqueda el primero el mejor.



Fig. 6. Simulador con amplitud.



Fig. 7. Árbol de búsqueda con poda.

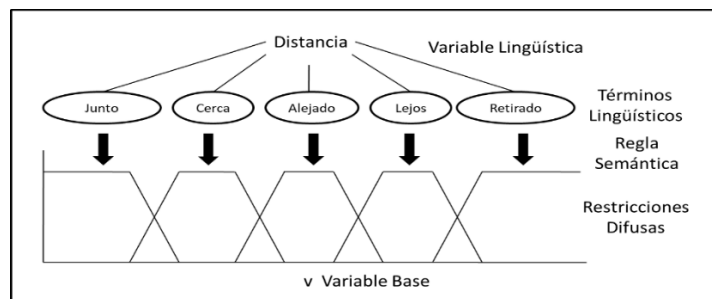


Fig. 8. Variable Lingüística distancia.

Como se puede observar en la Fig. 6 existen nodos que el árbol de búsqueda ha generado, los cuales se encuentran próximos a los objetos del mapa, han rodeado el perímetro de los objetos del mapa, para solucionar este problema se propone realizar una poda de los nodos, para lograr dicho proceso se formuló una función de membrecía triangular (Ecuación 2) cuyo objetivo es determinar los nodos que el robot puede explorar sin tener colisiones.

La Ecuación 2 es una función triangular, la cual tiene 4 casos, el primero  $x < a$ , la ecuación esta en  $0 \mu_A(x)$ , la segunda etapa es  $\frac{x-a}{b-a} a \leq x < b \mu_A(x)$  aqui se presenta la

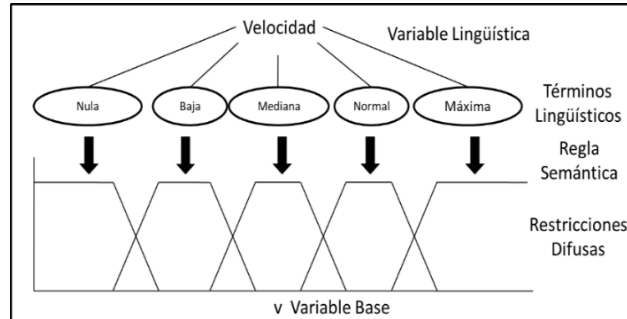


Fig. 9. Variable lingüística velocidad.

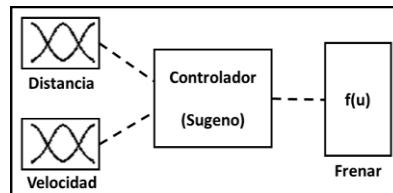


Fig. 10. Controlador difuso.

Tabla 1. Variables distancia, velocidad vs frenar.

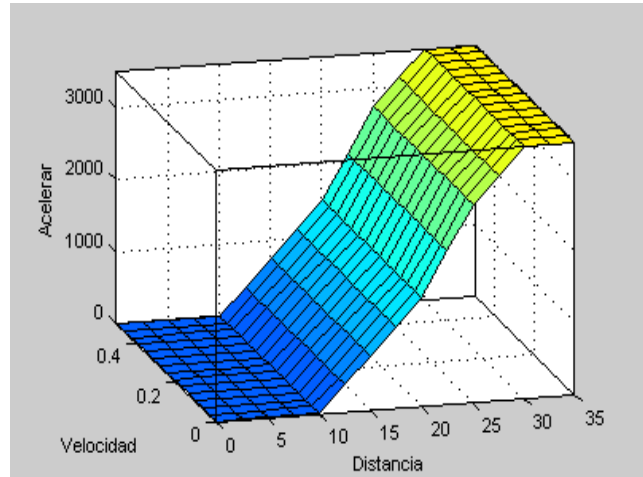
		Distancia				
		Junto	Cerca	Alejado	Lejos	Retirado
Velocidad	Nula	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Baja	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Media	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Alta	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Máxima	F(n)	F(m-)	F(m)	F(m+)	F(r)

pendiente de la función, la tercera etapa  $\frac{c-x}{c-b} \mathbf{b} \leq \mathbf{x} \leq \mathbf{c} \mu_A(\mathbf{x})$  es la pendiente que decrece de la función y la cuarta parte es cuando  $\mu_A(\mathbf{x})$  es *cero*.

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c < x \end{cases} \quad (2)$$

Al ser implementada la poda de los nodos, se puede observar en el simulador un espacio entre el árbol de búsqueda y los objetos del mapa como se observa en la Figura 7.

Para la implementación de la Lógica Difusa se realizó el procedimiento para obtener un par de conjuntos difusos, utilizando las variables independientes de velocidad y distancia, las cuales se utilizaron para ser transformadas en variables lingüísticas, las cuales son representadas en las Figuras 8 y 9.



**Fig. 11.** Curva de control.

**Tabla 2.** Reglas de control.

1. If ( <b>Distancia</b> is <b>Junto</b> ) and ( <b>Velocidad</b> is <b>Nula</b> )	then ( <b>Frenar</b> is <b>Nulo</b> ) (1)
2. If ( <b>Distancia</b> is <b>Junto</b> ) and ( <b>Velocidad</b> is <b>Baja</b> )	then ( <b>Frenar</b> is <b>Nulo</b> ) (1)
3. If ( <b>Distancia</b> is <b>Junto</b> ) and ( <b>Velocidad</b> is <b>Media</b> )	then ( <b>Frenar</b> is <b>Nulo</b> ) (1)
4. If ( <b>Distancia</b> is <b>Junto</b> ) and ( <b>Velocidad</b> is <b>Alta</b> )	then ( <b>Frenar</b> is <b>Nulo</b> ) (1)
5. If ( <b>Distancia</b> is <b>Junto</b> ) and ( <b>Velocidad</b> is <b>Maxima</b> )	then ( <b>Frenar</b> is <b>Nulo</b> ) (1)

Se diseñó el controlador difuso utilizando las variables de distancia y velocidad, al realizar este proceso se genera la variable dependiente *Frenar* (F), que es necesaria para realizar la exploración del móvil. (Fig. 10).

En la Tabla 1 se puede observar la interacción entre las variables distancia y velocidad con la variable *Frenar*, note las acciones de la última variable que son F(n): *Frenar* (nulo), F(m-): F(media-), F(m): F(media), F(m-): F(media-) y F(r): F(rápida).

Partiendo de la tabla anterior, se implementa el controlador difuso, describiendo las acciones de la variable *Distancia* con el valor de nula y los valores en la variable de *Velocidad* (nula, baja, media, alta, máxima). En la Tabla 2 se describen las acciones en la variable *Frenar*.

## 5. Resultados

En la Fig. 11 se muestra la respuesta del controlador difuso y la estabilidad del sistema, en la misma figura se puede analizar la interacción entre las variables independientes *Velocidad* y *Distancia*, con la variable dependiente *Frenar*. Al realizar el proceso del control difuso se logró el objetivo de este trabajo.



## 5.1 Algoritmo difuso

Para lograr el diseño del control difuso se formuló el nuevo algoritmo utilizando las reglas de control. A continuación, se describen las acciones de la variable Velocidad con el valor nulo y la variable Distancia.

### Algoritmo

#### 0. Inicio

1. Detectar distancia de los tres sensores.

I. Si central es igual a 0 rotar el móvil e ir a 3)

- |                                  |   |                             |   |
|----------------------------------|---|-----------------------------|---|
| 2. a) Si <b>velocidad</b> = nula | y | <b>distancia</b> = junto    | entonces <b>frenar</b> = nulo.          |
| b) Si <b>velocidad</b> = nula    | y | <b>distancia</b> = cerca    | entonces <b>frenar</b> = mediana menos. |
| c) Si <b>velocidad</b> = nula    | y | <b>distancia</b> = alejado  | entonces <b>frenar</b> = mediana.       |
| d) Si <b>velocidad</b> = nula    | y | <b>distancia</b> = lejos    | entonces <b>frenar</b> = mediana más.   |
| e) Si <b>velocidad</b> = nula    | y | <b>distancia</b> = retirado | entonces <b>frenar</b> = rápido.        |
| f) Si <b>velocidad</b> = baja    | y | <b>distancia</b> = junto    | entonces <b>frenar</b> = nulo.          |
| g) Si <b>velocidad</b> = baja    | y | <b>distancia</b> = cerca    | entonces <b>frenar</b> = mediana menos. |
| h) Si <b>velocidad</b> = baja    | y | <b>distancia</b> = alejado  | entonces <b>frenar</b> = mediana.       |
| i) Si <b>velocidad</b> = baja    | y | <b>distancia</b> = lejos    | entonces <b>frenar</b> = mediana más.   |
| j) Si <b>velocidad</b> = baja    | y | <b>distancia</b> = retirado | entonces <b>frenar</b> = rápido.        |
| k) Si <b>velocidad</b> = media   | y | <b>distancia</b> = junto    | entonces <b>frenar</b> = nulo.          |
| l) Si <b>velocidad</b> = media   | y | <b>distancia</b> = cerca    | entonces <b>frenar</b> = mediana menos. |
| m) Si <b>velocidad</b> = media   | y | <b>distancia</b> = alejado  | entonces <b>frenar</b> = mediana.       |
| n) Si <b>velocidad</b> = media   | y | <b>distancia</b> = lejos    | entonces <b>frenar</b> = mediana más.   |
| o) Si <b>velocidad</b> = media   | y | <b>distancia</b> = retirado | entonces <b>frenar</b> = rápido.        |
| p) Si <b>velocidad</b> = alta    | y | <b>distancia</b> = junto    | entonces <b>frenar</b> = nulo.          |
| q) Si <b>velocidad</b> = alta    | y | <b>distancia</b> = cerca    | entonces <b>frenar</b> = mediana menos. |
| r) Si <b>velocidad</b> = alta    | y | <b>distancia</b> = alejado  | entonces <b>frenar</b> = mediana.       |
| s) Si <b>velocidad</b> = alta    | y | <b>distancia</b> = lejos    | entonces <b>frenar</b> = mediana más.   |
| t) Si <b>velocidad</b> = alta    | y | <b>distancia</b> = retirado | entonces <b>frenar</b> = rápido.        |
| u) Si <b>velocidad</b> = máxima  | y | <b>distancia</b> = junto    | entonces <b>frenar</b> = nulo.          |
| v) Si <b>velocidad</b> = máxima  | y | <b>distancia</b> = cerca    | entonces <b>frenar</b> = mediana menos. |
| w) Si <b>velocidad</b> = máxima  | y | <b>distancia</b> = alejado  | entonces <b>frenar</b> = mediana.       |
| x) Si <b>velocidad</b> = máxima  | y | <b>distancia</b> = lejos    | entonces <b>frenar</b> = mediana más.   |
| y) Si <b>velocidad</b> = máxima  | y | <b>distancia</b> = retirado | entonces <b>frenar</b> = rápido.        |

3. Si no ha llegado a la meta ir a 1).

4. Salir.

## 6. Conclusiones

Resumamos, se estableció un par de conjuntos difusos con las variables independientes *distancia* y *velocidad*, al utilizarlas en el controlador difuso se logró obtener una variable dependiente *frenar*, además, se utilizaron las reglas del controlador difuso para formular un nuevo algoritmo. Resultando que la lógica difusa influye de manera importante para lograr la exploración de un mapa, ya que el controlador difuso es más rápido con respecto al largo proceso matemático del control tradicional.

## Referencias

1. McCarthy, J.: Padre de la Inteligencia Artificial. <https://www.bbvaopenmind.com/tecnologia/inteligencia-artificial> (1956)
2. Rifkin, J.: *The Third Industrial Revolution*. New York: Palgrave Macmillan (2011)
3. Pérez, G.G.: *Percepción y valores: Conceptos fecundos e incomprensidos*. Ciudad de México: Alternativas cubanas en psicología (2013)
4. Asker-Zadeh, L.A.: Fuzzy sets. *Information and Control*, pp. 338–353 (1965)
5. Morales-Jiménez, F.J.: Reporte de Robot Scribbler® (2012)
6. Robot Scribbler®. Parallax Inc. <https://www.parallax.com/product/28333> (2019)
7. Baturone, A.O.: *Robótica manipuladores y robots móviles*. Barcelona: Marcombo, S.A. (2001)
8. Moravec, H.P.: *Robots and Intelligence*. <https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1983/robage.mss> (1983)
9. John-Bares, W.: *cyberneticzoo.com*. <http://cyberneticzoo.com/walking-machines/1988-91-ambler-john-bares-william> (1988)
10. Durrant-Whyte, H., Rye, D., Nebot, E.: Localization of automatic guided vehicles. In: *The 7th International Symposium (ISRR'95)* (1995)
11. Corke, P.I., Trevelyan, J.: *DBLP computer science bibliography*. Sidney, Australia (1999)
12. Pons, J.V.: *Localización y generación de mapas del entorno (SLAM) de un robot por medio de una Kinect*. Valencia, España: Universidad Politécnica de Valencia (2012)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), pp. 116–132 (1985)
14. Marinelli, M., Kuna, H., Puente, F., Mounier, M., Kornuta, C., De Silvestre, E., Kornuta, C., Kolb, M.: *Aplicación de técnicas de inteligencia artificial para la navegación de robots móviles utilizando el simulador player/stage*. Universidad Nacional de Misiones Felix de Azara (2013)
15. Martín, F., Matellán, V., Barrera, P., Cañas, J.M.: *Localización basada en lógica difusa y filtros de Kalman para robots con patas*. Madrid España (2006)
16. Lozano, M.: *Aplicación de técnicas basadas en lógica difusa para la mejora del comportamiento de los algoritmos genéticos con codificación real*. Universidad de Granada (1997)
17. Parra, H., Ríos, L.: *Navegación de Robots móviles mediante comportamientos utilizando lógica difusa*. *Scientia et Technica*, Año XIII, 34, Universidad Tecnológica de Pereira, pp. 79–84 (2007)
18. Ibarra, M.: *Navegación autónoma de un robot con técnicas de localización y ruteo*. INAOE (2009)
19. Caballero, M., Cruz, L.: *Navegador Difuso de un Robot Movil*. <http://www.mecamex.net/anterior/cong02/papers/art29.pdf> (2019)
20. Costella, V., Rodríguez, H.: *Sistema de control con lógica difusa y diseño de plataforma autónoma con ruedas mecanum a partir de su arquitectura*. *RIDTEC*, 9(2), pp. 30–38 (2013)