NAVEGADOR DIFUSO DE UN ROBOT MOVIL

Marco A. Caballero, Luis Cruz, Ever A. Martínez y E. Gorrostieta
Instituto Tecnológico de Querétaro. Dep. Sistemas Computacionales.
Av. Tecnológico Esq. Escobedo Centro. C.P 76000 Santiago de Querétaro.
Querétaro.

RESUMEN

El continuo desarrollo de nuevos robots móviles para propósitos científicos y de aplicaciones diversas, exige algoritmos y técnicas de control eficiente para su desempeño. Este documento tiene como objetivo presentar el control de un robot móvil, haciendo uso de una de las técnicas más utilizadas en este sentido como es la Lógica Difusa. Dicha técnica es utilizada para controlar el avance del robot, y así cumplir su objetivo principal que es el llegar a un objetivo fijo de ubicación conocida. Dentro del espacio en que se mueve dicho robot, existen varios obstáculos que le impiden llegar a su meta. Es así como la Lógica Difusa le indica al robot que movimientos realizar en presencia de dichos obstáculos.

1. INTRODUCCION

A través de los años la ciencia y tecnología avanzan a pasos agigantados. A veces con mayor rapidez a veces con menos, sin embargo siempre sigue adelante. Y es con este avance que surgen nuevas ramas y espacios de aplicación. Uno de ellos y para este artículo, el más importante, es la rama de control aplicado a robots móviles.

Muchos son las técnicas y algoritmos de control para este tipo de robots, todos con características propias, cada una de ellas y ellos ofrecen ventajas y desventajas que permiten al científico la opción de utilizar el que más le sirva para su tarea o investigación.

Dada la especialización de nuevos mecanismos, se exigen así también técnicas de control capaces de ofrecer un trabajo satisfactorio en la realización de esta tarea.
Una de estas técnicas es la llamada "Lógica Difusa" (Borrosa, Fuzzy), la cual se considera una de las mejores. Esto es por su gran versatilidad, flexibilidad y sencillez. Dicha técnica se dio a conocer en los años 60 (En 1965 por Lotfi A. Zadeh, profesor de la Universidad de Berkeley California) con el articulo *"Fuzzy Sets" Information and Control.*

La fama de esta técnica, levantó rápidamente el vuelo gracias al uso que le dio Japón en 1987. La producción de un sinnúmero de productos japoneses aplicaba esta técnica obtenido resultados excelentes y superando claramente a sus contrincantes. Los productos que se producían y se siguen produciendo, y que utilizan esta técnica no se encasillan en un solo tipo; lavadoras, cámaras digitales, planchas entre muchos otros más, son los productos que hacen uso de ella. Esta aplicación le dio a Japón millones de ganancias anuales.

Pero no solo en este ambiente se desarrolla la Lógica Difusa. Muchas son las puertas que se han abierto para su aplicación. El ámbito científico, de investigación y de desarrollo son las principales áreas donde se aplica esta técnica con grandes resultados. La sencillez con que trabaja la lleva a ser una de las más entendidas y utilizadas, ya que ofrece excelentes resultados si es bien utilizada.

Tiempo, esfuerzo y dinero son algunas de las cosas que se invierten cuando se realiza una investigación, se desarrolla un nuevo invento o se mejora un experimento anterior. Años atrás, los científicos invirtieron esto y más para poder llevar a cabo sus experimentos. Las cantidades de dinero que se invertían en el desarrollo y manufacturación de lo que se desarrollaba, eran demasiado grandes.

Uno de los campos que ha tenido continuidad en su experimentación he implementación, es el diseño y desarrollo de robots móviles, y es la simulación la que ha permitido reducir considerablemente la inversión monetaria para su implementación y continua experimentación.

La simulación permite repetir una y otra vez un experimento, sin tener una inversión monetaria significativa. Es así como la simulación ha permitido el diseño y experimentación de una forma más sencilla y exitosa, sin tener que construir o modificar físicamente el experimento con que se está trabajando.

2. ROBOT MOVIL

Para propósitos de experimentación se utilizó un simulador hecho en OpenGL. Permitiendo una serie de repeticiones y experimentación para lograr el objetivo de este proyecto.

Las características de nuestro robot son las siguientes:

- Robot con base circular
- Dos llantas con motores independientes, encargados de la tracción de dicho robot.
- Una rueda loca.
- Tres sensores al frente: Uno central y dos mas separados a 30 grados, uno a la izquierda y otro a la derecha del central respectivamente.

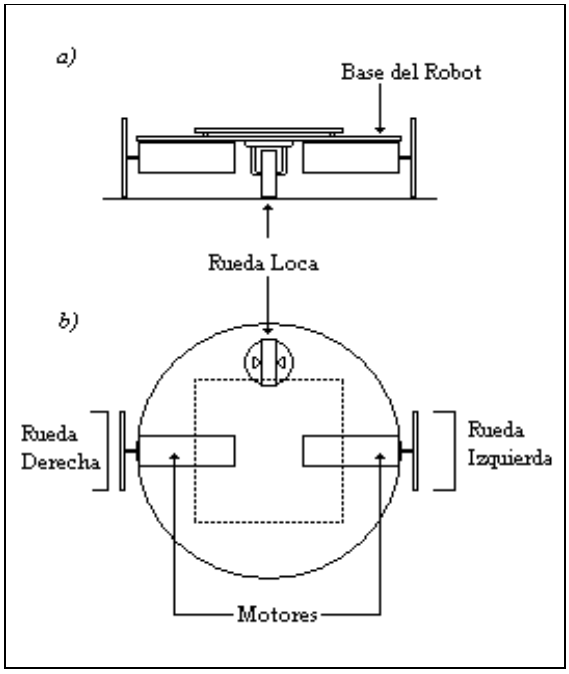

Figura 1: a) Vista frontal, b) Vista Aérea

Cabe mencionar nuevamente que el objetivo principal de este proyecto, es el controlar un robot con las características anteriormente mencionadas, mediante la Lógica Difusa. La implementación de esta técnica permitirá al robot tomar una serie de decisiones para poder llegar a un objetivo, en este caso una ubicación específica.

Las siguientes figuras muestran el robot desde diferentes perspectivas. Como se mencionó anteriormente se utilizó un simulador basado en OpenGL como herramienta:

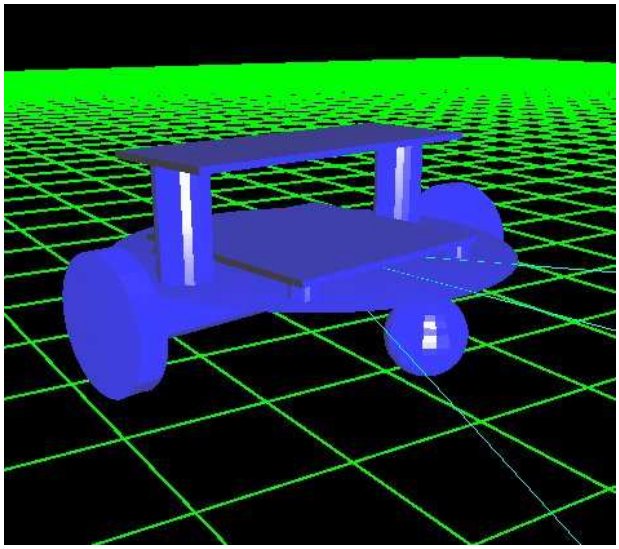

Figura 2. Vista ¾ de robot

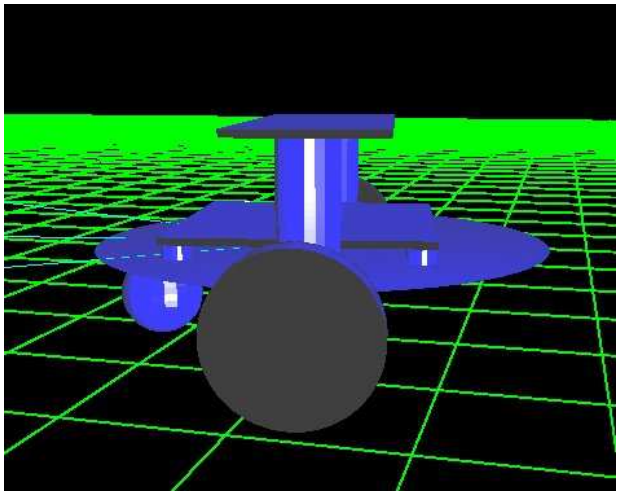

Figura 3. Vista lateral

El recorrido del robot se ve truncado por una serie de obstáculos, los cuales tendrán que ser librados, mediante los resultados que arroje la aplicación del algoritmo de la Lógica Difusa .

3 ALGORITMO DE LOGICA DIFUSA

Para lograr un control efectivo por medio de lógica difusa, dicho control debe tener ciertas características las cuales permitirán manipular las variables y el conocimiento con que se cuenta, así mismo manipular el robot.

Las características de un controlador difuso son:

- Base de conocimiento (“rule-base”)
- Mecanismo de inferencia (“inference mechanism”)
- Interfaz de difusificación (“fuzzyfication”)
- Interfaz de desdifusificación (“defuzzyfication”)

En forma general, el algoritmo de la lógica difusa se puede representar en la siguiente figura.

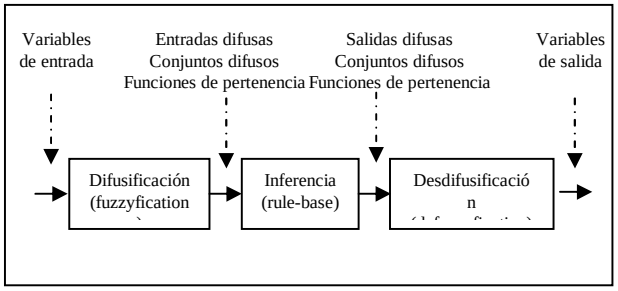

Figura 4: Estructura general del algoritmo de lógica difusa

3.1 Definición de variables de entrada y salida

Primero hay que definir las variables de entrada:

1.- Angulo relativo del robot con respecto al objetivo
2.- Distancia del robot al obstáculo

3.2 Definición conjuntos y valores lingüísticos asociados a las variables de entrada y salida

Ahora se definen los conjuntos y valores lingüísticos asociados a cada variable:

Para la variable “Angulo relativo” se definen los siguientes valores lingüísticos:

- Muy izquierda
- Izquierda
- Cero
- Derecha
- Muy derecha

Para la variable “Distancia al obstáculo” se definen los siguientes valores lingüísticos:

- Lejos izquierda
- Cerca izquierda
- Muy cerca izquierda
- Muy cerca derecha
- Cerca derecha
- Lejos derecha

3.3 Definición de funciones de pertenencia

Ahora para cada valor lingüístico de las dos variables se debe definir una función de pertenencia que indique el grado en que una variable “x” esta incluida en los conceptos representados por las variables lingüísticas.

Generalmente se utiliza $\mu_i(x)$ para indicar el grado en que “x” esta incluida dentro del conjunto “i”. La expresión $\mu_i(x)$ es conocida como la función de pertenencia de la variable **“x”** en el conjunto **“i”**. El valor de pertenencia de una variable dentro de un conjunto oscila entre los valores 0 a 1.

Ahora toda esta teoría debe ser migrada a datos reales. Estos datos son precisamente los datos que usamos en el desarrollo de este proyecto.

Variable “Angulo relativo”:

Definición de conjunto y valores lingüísticos:

Rango: $-180° \leq x < +180°$

Definición de funciones de pertenencia:

Muy izquierda:	$-180° \leq x < -80°$
Izquierda:	$-100° \leq x < -20°$
Cero:	$-40° \leq x < +40°$
Derecha:	$+20° \leq x < +100°$
Muy derecha:	$+80° \leq x < 180°$

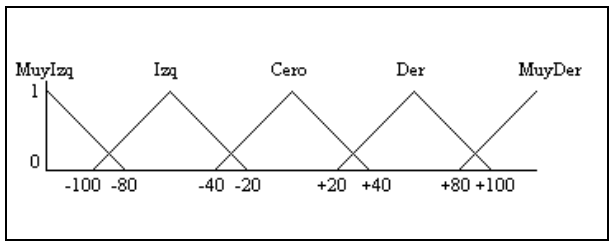

Figura 5: Variable de entrada ángulo relativo al objetivo

Variable “Distancia al obstáculo”:

Definición de conjunto y valores lingüísticos:

Rango: $-40\ u \leq x < +40\ u$

Definición de funciones de pertenencia:

Lejos izquierda:	$-40.0\ u \leq x < -26.8\ u$
Cerca izquierda:	$-37.2\ u \leq x < -5.0\ u$
Muy cerca izquierda:	$-21.2\ u \leq x < 0.0\ u$
Muy cerca derecha:	$0.0\ u \leq x < +21.2\ u$

Cerca derecha: +5.0 u ≤ x < +37.2 u
Lejos derecha: +26.8u ≤ x < +40.0 u

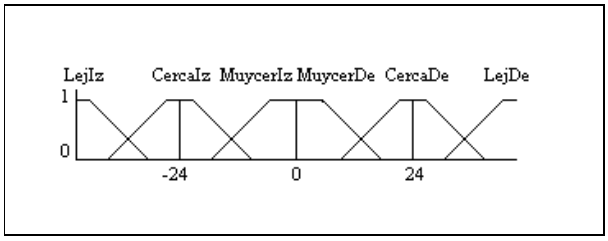

Figura 6: Variable de entrada de distancia al obstáculo

Es necesario también conocer las variables que tendremos de salida, en este caso el ángulo de giro que tendrá que realizar el robot.

Para la variable de salida "Angulo de giro" se definen los siguientes valores lingüísticos:

- Muy izquierda
- Izquierda
- Cero
- Derecha
- Muy derecha

Definición de conjunto y valores lingüísticos:

Rango: -30º ≤ x < 30º

Definición de funciones de pertenencia:

Muy izquierda: -30° ≤ x < -20°
Izquierda: -25° ≤ x < -5°
Cero: -10° ≤ x < +10°
Derecha: +5° ≤ x < +25°
Muy derecha: +20° ≤ x < +30°

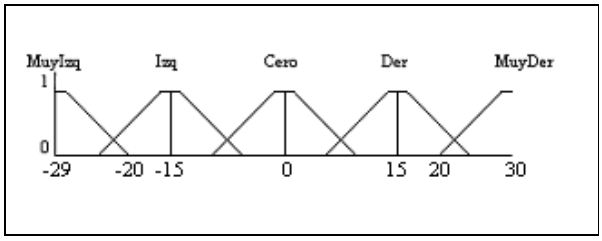

Figura 7: Variable de salida ángulo de giro

3.4 Operadores

Otro punto importante para el desarrollo del algoritmo de la Lógica Difusa, son los operadores.

Existen tres operadores básicos con los cuales se trabaja en este tipo de investigaciones: operador AND, operador OR y operador NOT.

Mientras que en la Lógica Clásica, los valores que se pueden obtener a partir de estas operaciones se limitan a 0 o 1, en la Lógica Difusa, dichos valores oscilan entre 0 y 1.

Definición de los operadores en la Lógica Clásica:

Intersección (AND):
min(A,B): $\mu_{A\cap B}(x)=\min[\mu_A(x),\mu_B(x)]$

Unión (OR):
máx(A,B):$\mu_{A\cdot B}(x)=\max[\mu_A(x),\mu_B(x)]$

Complemento (NOT): $\overline{\mu_A}(x)=1-\mu_A(x)$

A	B	AND	OR	NOT (A)	NOT (B)
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

Figura 8: Tabla con las operaciones AND, OR Y NOT aplicadas a las variables A y B

Definición de los operadores en la Lógica Difusa:

Intersección (AND):

Opción 1:
min(A,B): $\mu_{A\cap B}(x)=\min[\mu_A(x),\mu_B(x)]$

Opción 2:
prod (A,B) $\mu_{A\cap B}(x)=[\mu_A(x)\cdot\mu_B(x)]$

Unión (OR):
máx(A,B): $\mu_{A\cdot B}(x)=\max[\mu_A(x),\mu_B(x)]$

Complemento (NOT): $\overline{\mu_A}(x)=1-\mu_A(x)$

El operador utilizado para el desarrollo de este proyecto fue la Intersección (AND), Opción 1.

min(A,B): $\mu_{A\cap B}(x)=\min[\mu_A(x),\mu_B(x)]$

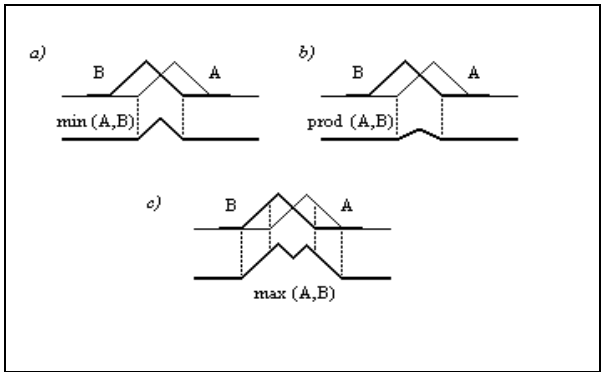

Figura 9: a) Operador AND Opción 1, b) Operador AND Opción 2, c) Operador OR

3.5 Reglas

Los conjuntos y operadores borrosos forman parte de frases, que posteriormente se utilizan para definir una acción u otra. Estas frases actúan dentro en el contexto de una sentencia del tipo if – then:

if "condiciones" then "acciones"

Ejemplo:

if $x_1$es y_1 and x_2 es y_2 then u_1 es w_1

Es de esta forma como se van formando una serie de reglas que rigen las acciones que realiza el robot. Estas reglas son concentradas en una tabla llamada BIOFAM (Binary Input-Output Fuzzy Asociative Memory).

Áng./Dis	LI	CI	MCI	MCD	CD	LD
MI	MI	MI	C	MI	MI	MI
I	MI	MI	C	MI	MI	I
C	C	D	MD	MI	I	C
D	D	MD	MD	C	MD	MD
MD	MD	MD	MD	C	MD	MD

Figura 10: BIOFAM. Giro contra "Angulo relativo" y "Distancia al obstáculo".

Para la figura anterior se consideran lo siguiente:

Áng. = Angulo relativo
Dis = Distancia al obstáculo
LI = Lejos izquierda
CI = Cerca izquierda
MCI = Muy cerca izquierda
CD = Cerca derecha
LD = Lejos derecha
MI = Muy izquierda
I = Izquierda
C = Cero
D = Derecha
MD = Muy derecha

3.6 Inferencia

"Se entiende por inferencia borrosa la interpretación de reglas if-then, con el objetivo de obtener *las conclusiones de las variables lingüísticas de salida* a partir de los *valores actuales de las variables lingüísticas de entrada*".[1]

Dentro de la etapa de inferencia podemos encontrar varios métodos para determinar el grado de pertenencia de las variables lingüísticas. Entre dichos métodos se pueden mencionar dos como los más importantes: Truncamiento y el Escalado.

Para propósito el desarrollo de este proyecto se utilizó el método de truncamiento.

Escalado:

$$\mu_{regla\_n}(u) = \mu_{premisa(n)} \cdot \mu_{acción(n)}(u)$$

Truncamiento:

$$\mu_{regla\_n}(u) = \text{mín}\ \{\mu_{premisa(n)}, \mu_{acción(n)}(u)\}$$

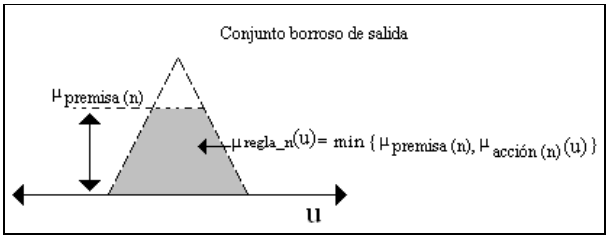

Figura 11: Esta figura muestra el conjunto borroso de salida que se obtiene a partir de utilizar el método de truncamiento

En la etapa de inferencia, la interpretación de las reglas if-then a partir de las variables lingüísticas es de suma importancia, puesto que de ello depende la siguiente etapa conocida como defusificación.

3.7 Defusificación

"La defusificación de las variables es su determinación como conjuntos difusos; representación fundamental para la posterior obtención de los valores de membresía en el proceso de evaluación de las reglas de inferencia."[2]

"La entrada para la defusificación es un conjunto borroso, el que resulta de la agregación, y la salida es un número (u_{crip})".[3]

Para llevar a cabo la defusificación existen varias opciones de las cuales destacan básicamente dos:

- Centro de gravedad (Este método es el que se utilizó en el desarrollo de este proyecto).
- Centros ponderados

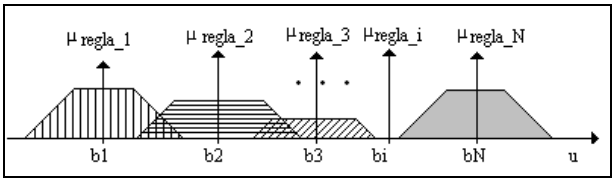

Figura 12: bi representa el centro de las funciones de pertenencia del conjunto para la regla "i", μ_{regla_i}

Este procedimiento incluye el seguimiento de la siguiente fórmula:

$$u^{crisp} = \frac{\Sigma_i b_i \int \mu_{regla\_i}}{\Sigma_i \int \mu_{regla\_i}}$$

Siendo $\int \mu_{regla\_i}$ el área bajo la función de pertenencia μ_{regla_i}.

4. **RESULTADOS.**

En las figuras se muestra la interfaz utilizada y los resultados obtenidos en el simulador con trayectorias realizadas satisfactoriamente.

Los obstáculos son representados por esferas dentro del simulador, las cuales pueden tener posiciones definidas o generadas aleatoriamente.

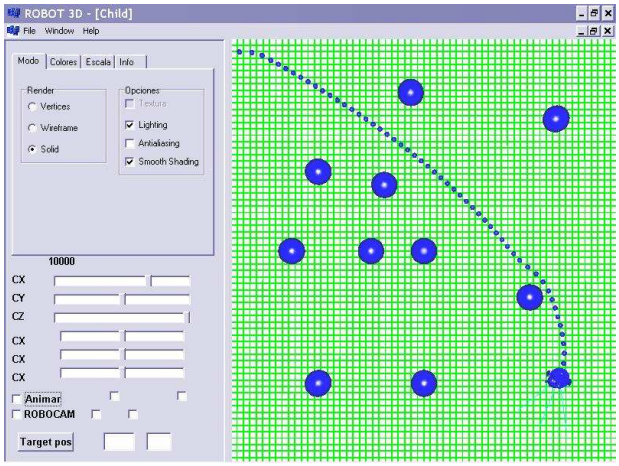

Figura 13. Trayectoria hecha por el robot.

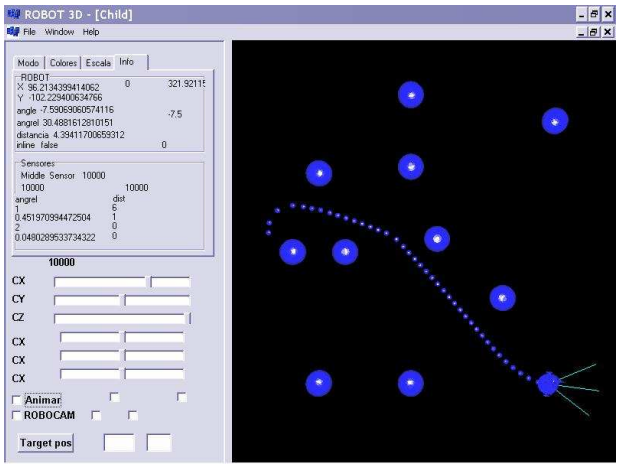

Figura 14. Trayectoria realizada por el robot con un mapa de obstáculos parecido al de la Fig.13 pero en un punto de partida del robot distinto.

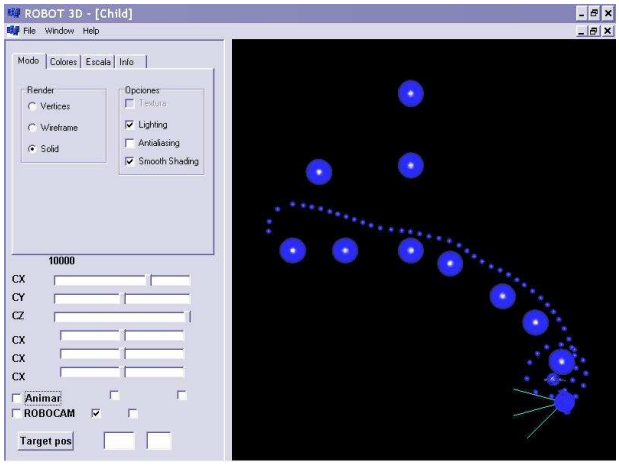

Figura 15**.** Comportamiento del robot al encontrar un obstáculo muy cercano al objetivo.

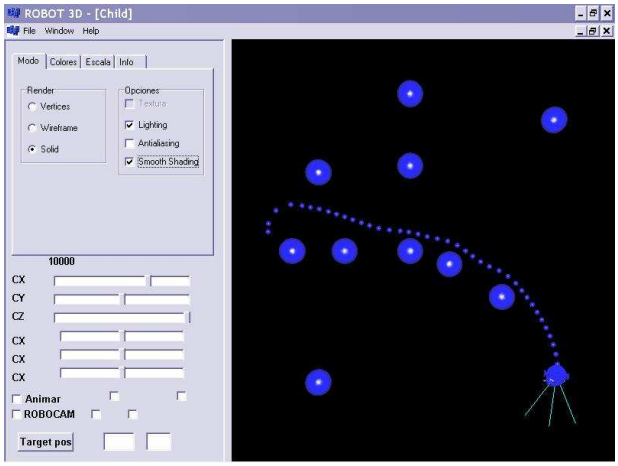

Figura 16. Trayectoria satisfactoria con un mapa de obstáculos similar al mostrado en la figura 15 esta vez sin obstáculo muy cercano al objetivo.

5. CONCLUSIONES

El robot realizó un acierto del 99% en las pruebas. Sin embargo, el robot no siempre sigue una trayectoria óptima. Esto puede corregirse agregando más reglas al modelo difuso y un algoritmo de optimización de trayectorias.

6. REFERENCIAS

[1] Berkan Riza C. and Trubatch Sheldon. , "Fuzzy System Design Principle ", IEEE Press. USA. 1997.

[2] Tanaka Kazuo., "An Introduction to Fuzzy Logic for Practical Applications ", Springer USA.1997.

[3] Manuel Mazo Quintas, "Control Borroso", Departamento de Electrónica. Universidad de Alcalá.

[4] Pedrycz Witold., "Fuzzy Control and Fuzzy Systems", Research Studies Press and John Wiley and Son Inc. London England. 1993.

[5] Angeles Jorge., "Fundamentals of Robotic Mechanical System", Springer USA.1997.

[6] Sarareh Babvey, Omid Mortahan, Mohammad R. Megbodi., "A Fuzzy Based Intelligent Navigation System for the Mobile Robot ", Proceeding of the 3th International Symposium on Robotics and Automation. Toluca Mexico. 2002.

[7] Vidal Calleja T.A, Velazco Villa and Aranda Bricarie E.,"Real Time Obstacle Avoidance For TrailerLike Systems-". Proceeding of the 3th International Symposium on Robotics and Automation. Toluca Mexico. 2002.

[8] Woo Mason, Neider Jakie, Davies Tom and Shreiner Dave. "The Official Guide to Learning OpenGL", 3rd, Addison Wesley Pub Co.USA 1999.

[9] Richard S., Whight Jr. and Sweet Michel R., "OpenGL Super Bible". Second Edition. Waite Group. USA. 1999.

[10] Segovia Armando, Garduño Mayra, Martinez Luis y Cruz Roberto., "Interfaz Grafica para el Control Interactivo de un Robot Movil", Segundo Congreso Nacional de Robótica, Asociación Mexicana de Robótica Toluca México 2000.

[11]Rene de la Calleja M. and Sánchez Abraham., "A motion Planer for Car like Robot ", In Proceedings of International Symposium on Robotics and Automation ISRA'2000 (Monterrey N. L. , Mexico, Nov 10–12).2000.