

# Control de lógica difusa

LUIS ALBERTO BALLADO ARADIAS

Cinvestav Unidad Tamaulipas

luis.ballado@cinvestav.mx

30 de abril de 2023

## Resumen

*El presente trabajo describe la implementación de un control de lógica borrosa (lógica difusa, fuzzy logic) en un robot móvil no holonómico LEGO NXT, bajo el lenguaje NXC (Not eXactly C). El control difuso es una técnica de control que se utiliza para sistemas que no tienen una modelización precisa o que son difíciles de modelar matemáticamente. Esta técnica permite trabajar con conceptos imprecisos y valores que no son necesariamente verdaderos o falsos, sino que tienen cierto grado de verdad o falsedad. Éste método depende mucho de un grado de experiencia de lo que se quiere controlar.*

## I. INTRODUCCIÓN

EL control difuso utiliza reglas lingüísticas para describir el comportamiento del sistema y para tomar decisiones de control. Estas reglas se basan en el conocimiento experto y la experiencia. Se expresan en términos lingüísticos, en lugar de expresiones matemáticas precisas.

El objetivo del control difuso es producir una salida que sea óptima en relación con las entradas del sistema, incluso si las condiciones de entrada cambian o si el sistema es complejo o no lineal. Esta técnica ha sido utilizada en una amplia variedad de aplicaciones, desde el control de robots y sistemas de control de procesos, hasta la gestión de tráfico y la toma de decisiones en sistemas de inteligencia artificial.

El control difuso se basa en el uso de funciones de pertenencia para representar el grado de pertenencia de un valor a un conjunto difuso. Estas funciones pueden ser triangulares, trapezoidales, gaussianas, entre otras, y se utilizan para describir el comportamiento del sistema en términos lingüísticos.

El proceso de control difuso consta de cuatro etapas principales: **la fuzzificación de las entradas, la evaluación de las reglas, la agregación de las salidas y la defuzzificación de la salida.**

En la etapa de **fuzzificación**, las entradas del sistema se transforman en valores difusos utilizando las funciones de pertenencia.

En la **evaluación de las reglas**, se utilizan las reglas lingüísticas para determinar la contribución de cada regla al resultado final.

En la **agregación de las salidas**, se combinan las salidas de las reglas para obtener una salida global.

---

Finalmente, en la etapa de **defuzzificación**, se transforma la salida difusa en un valor numérico para controlar el sistema.

Una de las ventajas del control difuso es su capacidad para manejar sistemas no lineales, imprecisos o inciertos, lo que lo hace adecuado para una amplia gama de aplicaciones. Sin embargo, también presenta algunas limitaciones, como la necesidad de un conocimiento experto para establecer las reglas lingüísticas y la complejidad en la selección y diseño de las funciones de pertenencia.

En general, el control difuso ha demostrado ser una técnica efectiva y flexible para el control de sistemas complejos, y continúa siendo objeto de investigación y desarrollo en la actualidad.

### I. Control difuso tipo Takagi-Sugeno

Este tipo de control difuso se utiliza para sistemas no lineales y se basa en la lógica difusa y el modelo Takagi-Sugeno. En este tipo de control, las entradas se transforman en valores lingüísticos mediante funciones de pertenencia, pero en lugar de reglas difusas, se utilizan modelos lineales y no lineales para generar la salida. El resultado final es una combinación de varios modelos, cada uno de los cuales se activa en función de la entrada.

### II. Control difuso tipo Mamdani

Es el tipo de control difuso más utilizado y se basa en la lógica difusa para generar salidas difusas. En este tipo de control, las entradas se transforman en valores lingüísticos mediante funciones de pertenencia y se definen las reglas difusas para cada variable de entrada y salida. Luego, se utiliza la inferencia difusa para generar una salida difusa y, finalmente, se realiza la defuzzificación para obtener una salida concreta.

### III. Pasos para efectuar un control difuso

1. **Definir** las variables lingüísticas de entrada y salida (sus rangos numéricos y sus funciones de pertenencia)
2. **Construir la base de reglas** que representa la estrategia de control
3. **Fusificar** los valores de entrada, hallar las funciones de pertenencia y evaluar cada una de las reglas.
4. **Razonamiento** ejecutar la **inferencia** para determinar la fuerza de las reglas activadas.
5. **Defusificar** usando el centroide de gravedad para determinar la acción a ser ejecutada

Recordemos que la lógica difusa se trabaja para aquellos problemas donde la complejidad de sus variables y manipulación son muy altas, de tal manera que no podemos usar algoritmos genéticos, ni redes neuronales por que el problema se volvería no calculable en un tiempo de espera corto y se volvería un problema NP (Tiempo No Polinomial), siendo eso lo que no buscaríamos al momento de controlar un robot, queremos una solución muy rápida y ágil para este problema.

**LÓGICA DIFUSA  $\Rightarrow$  LÓGICA DEL HOMBRE**

## II. CONTROL DIFUSO EN LA NAVEGACIÓN DE UN ROBOT MÓVIL LEGO

Sea la planta el ROBOT a controlar, donde  $x_f, y_f, \theta_f$  son la entrada de control y  $x, y, \theta$  las posiciones finales a las que queremos llegar. La figura 1 representa el comportamiento del sistema a controlar

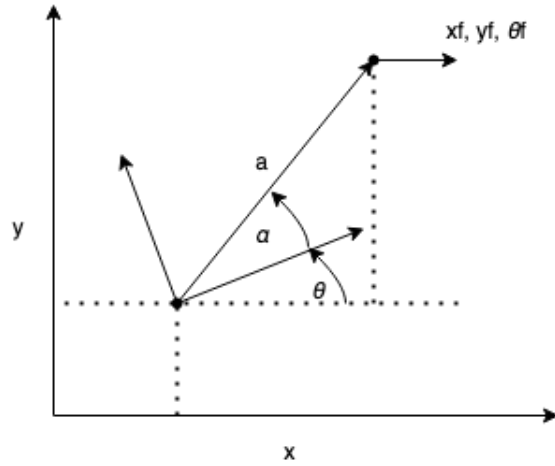


Figura 1: Sistema

Así como el control no lineal, en éste paradigma también buscamos el control de nuestro robot. Por lo que el sistema también se puede representar con el mismo diagrama.

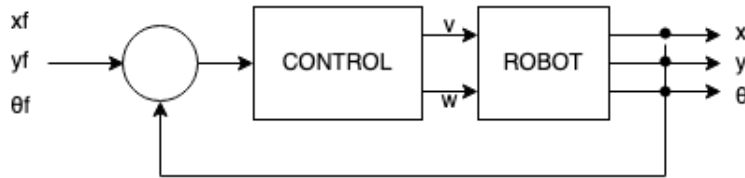


Figura 2: Sistema de Control Difuso

### I. Definición de variables lingüísticas

Nuestro dominio se muestra a continuación:

- Control de motores izquierdo (VL), derecho (VR)  $VL \in (-100, 100), VR \in (-100, 100)$ .
- Distancia de recorrido es variable (depende de  $x_f$  y  $y_f$ ),  $a \in [0, \infty)$
- Ángulo de error  $\alpha \in (-\pi, \pi)$

### II. Construcción de base de reglas

Se controlan 3 variables de salida.

Se parte de las reglas del experto. En esta práctica también se hace uso de la integración de odometría de prácticas pasadas, para obtener la distancia recorrida por nuestro robot no holonómico

en todo instante.

Construcción de la base de conocimientos

Para nuestro motor derecho se tiene:

	Z	C	L	ML
ID	0	-10	-10	-10
I	0	-10	-10	-10
Z	0	10	20	30
D	0	10	10	10
DD	0	10	10	10

Para nuestro motor izquierdo se tiene:

	Z	C	L	ML
ID	0	10	10	10
I	0	10	10	10
Z	0	10	20	30
D	0	-10	-10	-10
DD	0	-10	-10	-10

Las potencias de los motores se pueden ajustar para tener un mejor desempeño, en nuestro caso se tiene un comportamiento lento que se puede corregir aumentando los numeros en el rango de dominio de los motores.

### III. Fuzzificar, inferencia, defusificar

Las membrecias se definen como el gráfico de distancia.

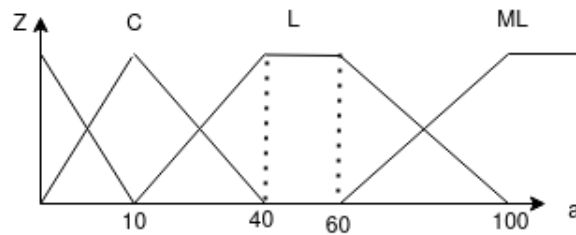


Figura 3: Distancia

- Z - Zero
- C - Cerca
- L - Lejos
- ML - Muy Lejos

y su inclinación respecto al  $x_f, y_f$  destino.

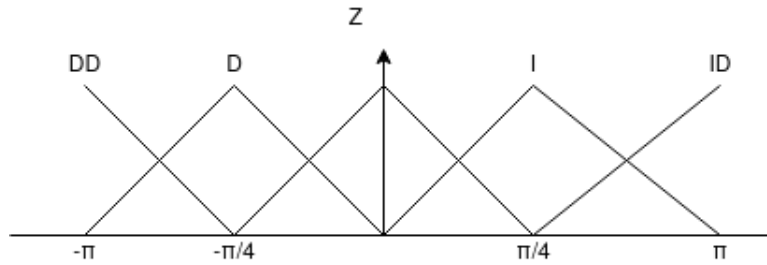


Figura 4: Orientación

- ID - Izquierda Detrás
- I - Izquierda
- Z - Zero (de frente)
- D - Derecha
- DD - Derecha Detrás

Apartir de los conocimientos vertidos en se puede obtener la distancia para cada instante respecto al desplazamiento del robot:

```

if  $a < 10$  then
   $z \leftarrow 1 - (a/10);$ 
   $c \leftarrow a/10;$ 
   $l \leftarrow 0; ml \leftarrow 0;$ 
else if  $a < 50$  then
   $z \leftarrow 0;$ 
   $c \leftarrow 1 - ((a - 10)/40);$ 
   $l \leftarrow ((a - 10)/40); ml \leftarrow 0;$ 
else if  $a < 100$  then
   $z \leftarrow 0;$ 
   $c \leftarrow 0;$ 
   $l \leftarrow (1 - ((a - 50)/50)); ml \leftarrow ((a - 50)/50);$ 
else
   $z \leftarrow 0;$ 
   $c \leftarrow 0;$ 
   $l \leftarrow 0; ml = 1;$ 
end

```

Algorithm 1: Desplazamiento

---

Para la inclinación  $DD \leftarrow -\pi$ ,  $D \leftarrow \frac{-\pi}{4}$ ,  $Z \leftarrow 0$ ,  $I \leftarrow \frac{\pi}{4}$ ,  $DI \leftarrow \pi$ :

```
if  $\alpha < D$  then
   $MD \leftarrow \frac{\alpha - D}{D - DD}$ ;
   $MDD \leftarrow 1 - MD$ ;
   $MZ \leftarrow 0$ ;
   $MI \leftarrow 0$ ;
   $MDI \leftarrow 0$ ;
else if  $\alpha < Z$  then
   $MZ \leftarrow \frac{\alpha - D}{Z - D}$ ;
   $MD \leftarrow 1 - MZ$ ;
   $MDD \leftarrow 0$ ;
   $MI \leftarrow 0$ ;
   $MDI \leftarrow 0$ ;
else if  $\alpha < I$  then
   $MI \leftarrow \frac{\alpha - Z}{I - Z}$ ;
   $MZ \leftarrow 1 - MI$ ;
   $MDD \leftarrow 0$ ;
   $MD \leftarrow 0$ ;
   $MDI \leftarrow 0$ ;
else
   $MDI \leftarrow \frac{\alpha - I}{DI - I}$ ;
   $MI \leftarrow 1 - MDI$ ;
   $MDD \leftarrow 0$ ;
   $MD \leftarrow 0$ ;
   $MZ \leftarrow 0$ ;
end
```

**Algorithm 2:** Inclinación

---

Apartir de los valores obtenidos de distancia e inclinación se calcula la potencia de los motores

```
for ( i = 0; i < 4; i = i + + ) {  
    for ( j = 0; j < 5; j = j + + ) {  
        if MA[i] < MAlpha[j] then  
            coef = MA[i];  
        else  
            coef = MAlpha[j];  
        acc_coef = acc_coef + coef;  
        acc_accion = acc_accion + coef * VL[j][i];  
    }  
}  
accion_VL =  $\frac{acc\_accion}{acc\_coef}$ 
```

**Algorithm 3:** Potencia para motor Izquierdo

```
for ( i = 0; i < 4; i = i + + ) {  
    for ( j = 0; j < 5; j = j + + ) {  
        coef2 = min(MA[i], MAlpha[j]);  
        acc_coef2 = acc_coef2 + coef2;  
        acc_accion2 = acc_accion2 + coef2 * VR[j][i];  
    }  
}  
accion_VR =  $\frac{acc\_accion2}{acc\_coef2}$ 
```

**Algorithm 4:** Potencia para motor Derecho

---

### III. RESULTADOS

El comportamiento que se obtuvo representó un desplazamiento como el siguiente gráfico. Mostrando la corrección de  $\alpha$  desde su inicio, para comenzar la navegación.

Se puede observar un pequeño video de la demostración del trabajo. [ver video](#)

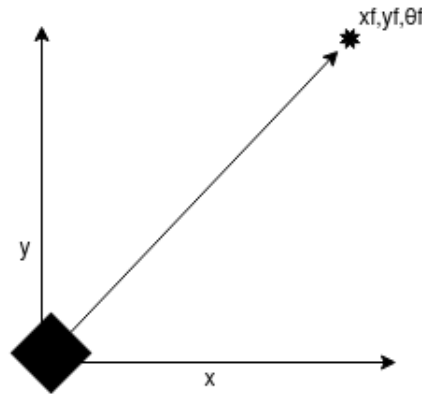


Figura 5: Desplazamiento del robot móvil

Nuevamente, nuestro robot puede presentar errores sistematicos como derrapes, u obstáculos dentro de nuestro camino. Así como potencias bajas en los motores apreciando un desplazamiento nulo, este último se puede corregir reasignando nuevos valores a nuestra matriz de conocimientos.

[Ver códigos fuentes github](#)

### IV. CONCLUSIONES

La implementación del un control difuso nos brinda grandes ventajas, pero solo es interpolador. Fuera de los rangos a controlar no obtendremos una solución.

En conclusión, el control difuso tipo Sugeno es una técnica de control basada en la lógica difusa que se diferencia del control difuso tipo Mamdani en que las reglas difusas definen funciones lineales en lugar de funciones de pertenencia. En este tipo de control, las entradas se transforman en valores lingüísticos mediante funciones de pertenencia, y se definen reglas difusas que se traducen en funciones lineales para cada variable de salida.

El control difuso Sugeno es útil en aplicaciones donde la precisión es fundamental y las variables de entrada tienen una relación lineal con la salida. Su capacidad para modelar sistemas no lineales de manera más precisa que el control difuso Mamdani lo hace adecuado para aplicaciones que requieren una alta precisión, como en el control de procesos industriales.

Sin embargo, el control difuso Sugeno puede resultar más difícil de diseñar que el control difuso Mamdani debido a la necesidad de definir funciones lineales para cada variable de salida. Además, la interpretación de los resultados puede resultar más difícil debido a la naturaleza lineal de las funciones de salida.

En resumen, el control difuso Sugeno es una técnica de control poderosa y efectiva en aplicaciones donde la precisión es fundamental y las variables de entrada tienen una relación lineal con la salida. Aunque puede presentar algunas dificultades en el diseño y la interpretación, su



---

capacidad para modelar sistemas no lineales de manera más precisa lo hace una herramienta valiosa en el campo del control automático.

## REFERENCIAS

- [1] Intalación NXC en LINUX, <http://ubuntudaily.blogspot.com/2011/03/using-lego-mindstorms-nxt-with-ubuntu.html>
- [2] Repositorio Compilador NBC utilizado, <https://github.com/pierre-24/nbc-compiler>
- [3] Documento para evitar sudo en NXC, <https://bricxcc.sourceforge.net/nbc/doc/nxtlinux.txt>
- [4] Comando para instalar libusb-dev, <https://howtoinstall.co/en/libusb-dev>
- [5] Presentación Odometría, <http://www.kramirez.net/Robotica/Material/Presentaciones/Odometria.pdf>
- [6] Modelo Cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odométrica, <https://www.redalyc.org/pdf/849/84916680034.pdf> VALENCIA V., JHONNY A.; MONTOYA O., ALEJANDRO; RIOS, LUIS HERNANDO
- [7] Modelo cinemático de un robot móvil implementado con LEGO NXT para un sistema de localización indoor diseñado en Labview, [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwio0\\_So28\\_9AhViDkQIHR1XDtcQFnoECBQQAQ&url=https%3A%2F%2Frevistas.udistrital.edu.co%2Findex.php%2FTecnura%2Farticle%2Fdownload%2F6810%2F8394%2F30717&usg=A0vVaw2PsCrkFGkk\\_nGN-G084B11](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwio0_So28_9AhViDkQIHR1XDtcQFnoECBQQAQ&url=https%3A%2F%2Frevistas.udistrital.edu.co%2Findex.php%2FTecnura%2Farticle%2Fdownload%2F6810%2F8394%2F30717&usg=A0vVaw2PsCrkFGkk_nGN-G084B11)
- [8] Notas de clase, Robótica Móvil Inteligente, Dr. José Gabriel Ramírez Torres, Enero-Abril 2023