



Universidad Tecnológica de Bolívar

Proyecto: Robot Mini Sumo con Control Fuzzy

Control Inteligente

Integrantes del grupo:

Daniel Yepes Tapia – T00058255

Jhonny Stevenson Seña – T00046169

Carmen Daniela Díaz Durango – T00062151

Ángel Carrascal – T00061068.

Héctor Marrugo – T00054717.

## **1. INTRODUCCIÓN**

En el campo de la ingeniería mecatrónica, la creación de un mini robot de sumo de competición ofrece valiosas oportunidades de aprendizaje y aplicación práctica en diversas disciplinas. Este tipo de proyecto integra mecánica, diseño 3D, electrónica, programación y sistemas de control, lo que permite a los estudiantes desarrollar y aplicar competencias interdisciplinarias, la esencia de este proyecto radica en el desarrollo de un mini robot de sumo que no sólo cumpla los requisitos estructurales y operativos básicos, como ser compacto y ágil en un espacio reducido, sino que además utilice técnicas avanzadas de control inteligente para optimizar su rendimiento en combate. En concreto, se busca un sistema de control difuso para mejorar la toma de decisiones del robot durante los combates.

El control difuso proporciona una forma eficaz de manejar la incertidumbre y la variabilidad inherentes a un mini combate de sumo. Utilizando lógica difusa, el robot puede adaptar sus estrategias de lucha en tiempo real y ser más sensible a diferentes situaciones. Esto incluye la selección dinámica de técnicas de ataque y defensa, optimizando así el rendimiento del robot en el ring.

Adicionalmente este enfoque no sólo avanza en la comprensión y aplicación de técnicas de control inteligente en robótica, sino que también proporciona a los estudiantes experiencia práctica en el diseño y optimización de sistemas complejos.

## **2. OBJETIVO GENERAL**

Implementar un sistema de control difuso (fuzzy) para optimizar las estrategias de combate de un robot mini sumo en competencias.

### **2.1 Objetivos específicos**

- Desarrollar un modelo de control difuso que permita la toma de decisiones estratégicas basadas en la interacción dinámica entre el robot y su oponente en el ring.
- Programar el controlador difuso en el entorno de desarrollo adecuado y asegurar que pueda ajustar dinámicamente las estrategias de combate del robot según las condiciones cambiantes del combate.
- Ajustar el sistema de control difuso a través de pruebas experimentales en el ring para mejorar su efectividad en la toma de decisiones durante los combates.
- Documentar el proceso de desarrollo del sistema de control difuso y sus impactos en el desempeño del robot.

### **3. ANTECEDENTES.**

Los campeonatos de robot sumo tienen más de 25 años a nivel mundial y todo comenzó en Japón a finales de los años 80. Estos campeonatos fueron creados con la finalidad de incentivar a los estudiantes a interesarse en la robótica, permitiéndoles adquirir experiencia y habilidad en la construcción de robots. En Ecuador, la Escuela Politécnica del Litoral (ESPOL) fue sede del primer concurso de robótica ecuatoriano en 2006, conocido como el Concurso Ecuatoriano de Robótica (CER). La Universidad Católica de Santiago de Guayaquil (UCSG) participó por primera vez en el CER en 2014, año en el que también se creó el club de robótica en esta universidad, con participación en la categoría mini sumo y otras categorías.

La robótica móvil ofrece una plataforma atractiva para combinar sistemas mecánicos, electrónicos, informáticos y de control, creando un sistema integrado para la educación y la investigación. Los robots móviles son dispositivos mecánicos capaces de moverse en un entorno con cierto grado de autonomía, utilizando sensores externos para capturar información del entorno. La autonomía de movimiento en robótica se asocia con la capacidad de realizar movimientos sin intervención externa, lo cual depende del rendimiento de los sensores y del algoritmo de control.

El controlador de lógica difusa (FLC) se utiliza como método de control no lineal para el razonamiento de las salidas de los sensores y la navegación del robot. La lógica difusa es un método simple de fusión de sensores para combinar las salidas de los sensores. En el contexto de la competencia de robots sumo, el objetivo principal es forzar al competidor a salir del ring mientras se permanece dentro del mismo. Este trabajo sugiere una solución FLC para la detección y seguimiento del oponente, utilizando sensores infrarrojos (IR) para la detección del objetivo. El algoritmo de control desarrollado muestra un mejor rendimiento en comparación con los algoritmos tradicionales de robots sumo, detectando y atacando al oponente en menos tiempo.

La robótica móvil ha avanzado significativamente, permitiendo la creación de sistemas autónomos capaces de navegar en entornos desconocidos y evitar obstáculos sin intervención humana. Este progreso se ha logrado mediante la integración de sensores, algoritmos de control y actuadores, que juntos permiten a los robots tomar decisiones en tiempo real.

En particular, la lógica difusa ha demostrado ser una herramienta eficaz para el desarrollo de sistemas de evasión de obstáculos. A diferencia de los métodos tradicionales basados en geometría y medición de distancias, la lógica difusa ofrece una mayor tolerancia a la imprecisión y la incertidumbre, lo que la hace ideal para aplicaciones en robótica móvil.

El artículo se centra en el desarrollo de un robot ligero de dos ruedas equipado con un sensor ultrasónico y controlado por un algoritmo de lógica difusa.

## **4. ESPECIFICACIONES TÉCNICAS**

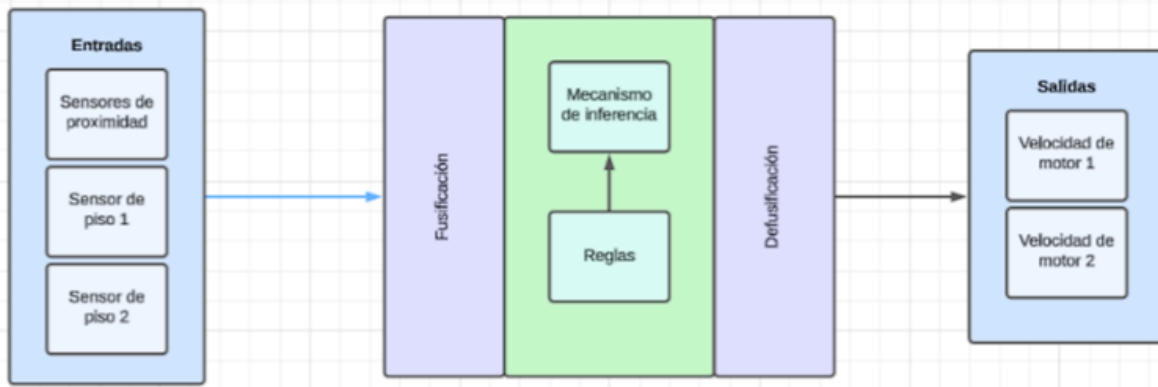
### **4.1. Componentes principales**

- Microcontrolador: Arduino Nano
- Sensores de distancia: 4-5 (a seleccionar)
- Sensores de piso: 2
- Controlador de motores: Compatible con Arduino Nano
- Motores: 2 motores DC con suficiente torque para competencias de mini sumo
- Fuente reguladora de voltaje.
- Circuito impreso o hecho con váquela perforada.
- Chasis diseñado e impreso en 3D.
- Cuchilla frontal.
- Base de acero.
- Tornillería.
- Batería de 12V.

### **4.2. Sistema de control**

Se implementará un sistema de control fuzzy para procesar las entradas de los sensores y determinar las acciones del robot. Este sistema permitirá una toma de decisiones más suave y adaptativa en comparación con un sistema de control tradicional.

## 5. MODELO FUZZY



*Ilustración 1: Modelo de sistema de control difuso*

Como se puede apreciar en la ilustración 1, se utilizará un sistema de control difuso que comenzará con las entradas de los sensores y el error, pasará por la fusificación, para pasar al mecanismo de inferencia con las reglas, luego a la defusificación y terminar mandando los sistemas de comandos al robot.

### 5.1. Entradas del sistema:

- Sensores de Proximidad: Detectan la distancia al oponente y a los bordes del ring, ej: sensor IR.

### 5.2. Bloque de fusificación:

Convierte las lecturas de los sensores en valores difusos.

### 5.3. Ejemplos de Reglas:

- Si el robot está cerca del oponente y se mueve rápido, entonces atacar con alta intensidad.

#### 5.4. Mecanismo de Inferencia:

Combina las reglas de acuerdo con las entradas difusas para obtener una salida difusa.

#### 5.5. Bloque de Defuzzificación:

Convierte la salida difusa en comandos precisos para el robot.

#### 5.6. Sistema de comandos:

**Comandos de movimiento:** Son los comandos que ajustan la velocidad y dirección del robot.

**Comandos de Estrategia:** Deciden la estrategia de ataque o defensa basada en la salida difusa.

Un ejemplo rápido para entender:

SENSOR	DESCRIPCIÓN	LECTURA (CM)	CATEGORÍAS DE PERTENENCIA
SENSO R 1	Proximidad delantera I	12	Cerca, Media, Lejos.
SENSO R 2	Proximidad delantera D	12	Cerca, Media, Lejos.
SENSO R 3	Proximidad izquierda	10	Cerca, Media, Lejos.
SENSO R 4	Proximidad derecha	10	Cerca, Media, Lejos.

Tabla de Funciones de Pertenencia

SENSOR	CATEGORÍA	RANGO DE LECTURA (CM)	FUNCIÓN DE PERTENENCIA
SENSOR 1-4	Cerca	0 - 5	Triangular o trapezoidal; alta membresía en 0 - 5 cm
SENSOR 1-4	Media	5 - 15	Triangular; baja membresía en 5 - 15 cm
SENSOR 1-4	Lejos	15 - 20	Triangular; baja membresía en 15 - 20 cm

Tabla de Inferencia

REGLA	ENTRADAS DIFUSAS	SALIDA DIFUSA
REGLA 1	Sensor 1 = Cerca, Sensor 2 = Lejos	Alta intensidad de avance
REGLA 2	Sensor 3 = Cerca, Sensor 4 = Lejos	Giro hacia la derecha
REGLA 3	Todos los sensores = Media	Mantener posición

**REGLA 4** | Sensor 1 = Lejos, Sensor 2 =      Giro hacia la izquierda  
Cerca

## 6. FASES DEL PROYECTO

### 6.1. Investigación y diseño (2 semanas)

- Investigar y seleccionar los sensores de distancia más adecuados.
- Diseñar el esquema eléctrico del robot.
- Diseñar el chasis del robot teniendo en cuenta las reglas de las competencias de mini sumo.

### 6.2. Implementación del hardware (3 semanas)

- Adquirir los componentes necesarios.
- Construir el chasis del robot.
- Montar los componentes electrónicos y realizar las conexiones.

### 6.3. Desarrollo del software (4 semanas)

- Implementar el sistema de control fuzzy en Arduino.
- Desarrollar algoritmos para la detección del oponente y los límites del ring.
- Crear estrategias de ataque y defensa

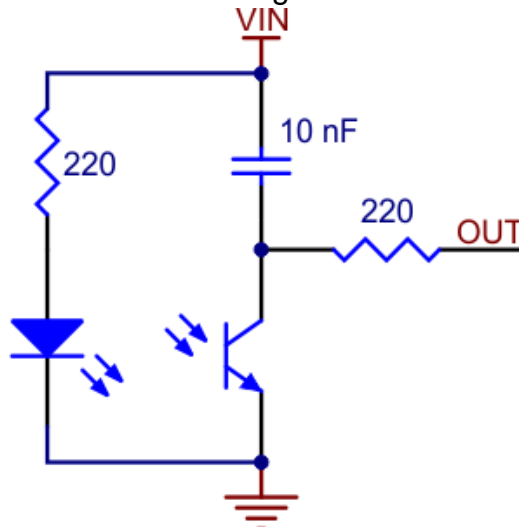
### 6.4. Pruebas y optimización (3 semanas)

- Realizar pruebas del robot en diferentes escenarios.
- Ajustar los parámetros del control fuzzy para mejorar el rendimiento.
- Optimizar el código para ajustarse de manera apropiada a las tácticas de combate deseadas.

## 7. LECTURA DE LOS SENSORES QTR-1C

Par empezar, tuve algunas dificultades para leer estos sensores. Debido a que vienen de etiqueta como digitales uno pensaría que sus salidas son 0 y 1.

El esquema electrónico del sensor es el siguiente:



Lo que lo diferencia del análogo es el condensador de 10nF.

El código lectura del sensor:

```
void setup(){
  Serial.begin(9600);
}

void loop(){
  int sensor_piso = Leer_Sensor_Piso (A0);
  Serial.println(sensor_piso);
  delay(5);
}

//////////FUNCION PARA LEER EL SENSOR DE PISO//////////
int Leer_Sensor_Piso (int pin){
  int PinState;
  int cnt;
  int sensor;
  pinMode (pin, OUTPUT);
  digitalWrite(pin, HIGH);
  delay(1);

  pinMode (pin, INPUT);

  cnt = 0;

  do{
    cnt ++;
    PinState = digitalRead(pin);
  }while (PinState== HIGH && cnt < 300);
  sensor = cnt;
  return sensor;
}
//////////
```

En este código se destaca principalmente la función de lectura del sensor qtr. Este código me entrega valores entre 0 y 300 con los que ya podría hacer una calibración dependiendo de la pista donde se encuentre el robot.

- Intente usar los pines A6 y A7 de mi pcb hecha a mano, pero no reconoce los valores. Tal parece que estos pines son para sensores digitales (Los míos son digitales, pero necesito una lectura análoga). Así que cambiare los pines a dos de pwm, en este caso serán el 3 y el 9.
- El pin 9 se encarga del sensor derecho y el 3 del izquierdo.

Por lo tanto, para leer ambos sensores de piso e imprimir los valores, creamos una nueva función de la siguiente manera:



```

////////FUNCION PARA LEER AMBOS SENSORES E IMPRIMIRLOS////////
int sensores_de_piso(){
  for(int i=0; i<2; i++){
    lectura_qtr[0] = Leer_Sensor_Piso (3); //sensor izquierdo
    lectura_qtr[1] = Leer_Piso (9); //sensor derecho
    Serial.print(lectura_qtr[i Sensor_]);
    Serial.print("\t");
  }
  Serial.println(" ");
}
////////

```

Es necesario que se declare un vector en el inicio del código de dos posiciones:

```
int lectura_qtr[2];
```

El valor del umbral lo dejare en una media de 150, es decir, que si es mayor a 150 el sensor me marcara como 0 y si es menos a 150 me marcara como 1. De esta forma podremos determinar la franja blanca.

Es necesario que se haga la conversión a digital, para ello modificamos el código anterior de la siguiente manera:

```

////////FUNCION PARA LEER AMBOS SENSORES E IMPRIMIRLOS////////
int sensores_de_piso(){
  for(int i=0; i<2; i++){
    lectura_qtr[0] = Leer_Sensor_Piso (3); //sensor izquierdo
    lectura_qtr[1] = Leer_Sensor_Piso (9); //sensor derecho

    //Conversion a digital
    if(linea==0){if(lectura_qtr[i]<=umbral){lectura_qtr[i]=0;}else{lectura_qtr[i]=1;}}
    if(linea==1){if(lectura_qtr[i]<=umbral){lectura_qtr[i]=1;}else{lectura_qtr[i]=0;}}

    Serial.print(lectura_qtr[i]);
    Serial.print("\t");
  }
  Serial.println(" ");
}
////////

```

En la parte superior declaramos el umbral:

```
int umbral = 150; //El umbral lo debemos promediar dependiendo de la pista en la que estemos
```

Y la variable línea que estará en cado de que en el domo se intercambien los colores. Para este caso lo dejamos en 0 porque la línea es blanca y el domo negro.

```

//////////COLOR DEL DOMO//////////
int linea=0; //dependiendo del color de domo, modificar esta variable
//////////

PROGRAMA FINAL PARA LA LECTURA DE LOS QTR

int lectura_qtr[2]; // en este vector se almacenan los datos de mis sensores de piso
int digital[2]; //guardaremos la conversion a digital segun el umbral
int umbral = 100; //El umbral lo debemos promediar dependiendo de la pista en la que estemos

//////////COLOR DEL DOMO//////////
int linea=0; //dependiendo del color de domo, modificar esta variable
//////////

void setup(){ Serial.begin(9600);
}

void loop(){ sensores_de_piso();
}

//////////FUNCION PARA LEER EL SENSOR DE PISO////////// int
Leer_Sensor_Piso (int pin){
    int PinState; int cnt;
    int sensor;
    pinMode (pin, OUTPUT);
    digitalWrite(pin, HIGH); delay(1);

    pinMode (pin, INPUT);

    cnt = 0;

    do{
        cnt ++;
        PinState = digitalRead(pin);
    }while (PinState == HIGH && cnt < 300); sensor = cnt;
    return sensor;
}
//////////

//////FUNCION PARA LEER AMBOS SENSORES E IMPRIMIRLOS//////////
int sensores_de_piso(){ for(int i=0;
    i<2; i++){
        lectura_qtr[0] = Leer_Sensor_Piso (3); //sensor izquierdo lectura_qtr[1] =
        Leer_Sensor_Piso (9); //sensor derecho
    }
}

```

```

//Conversion a digial
if(linea==0){if(lectura_qtr[i]<=umbral){lectura_qtr[i]=0;}else{lectura_qtr[
i]=1;}}
if(linea==1){if(lectura_qtr[i]<=umbral){lectura_qtr[i]=1;}else{lectura_qtr[
i]=0;}}

Serial.print(lectura_qtr[i]);
Serial.print("\t");
}
Serial.println(" ");
}

```

////////////////////////////////////

## 8. Morfología minisumo

El robot minisumo estará equipado con 5 sensores digitales infrarrojos distribuidos a lo largo de su chasis (ver Figura 1) para la detección del oponente (ver Figura 2). Estos sensores se ubicarán con una separación de 45°, cubriendo las direcciones: este, noreste, norte, noroeste y oeste, lo que proporcionará un rango de detección activo de 0 a 180 grados (ver Figura 3).

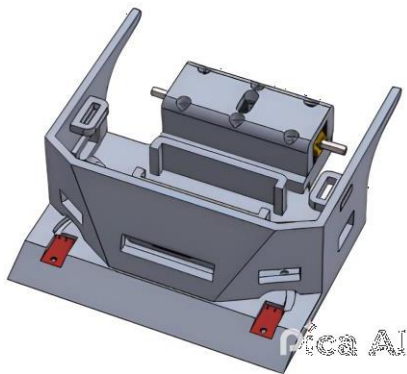


Figure 1. Chasis de ejemplo

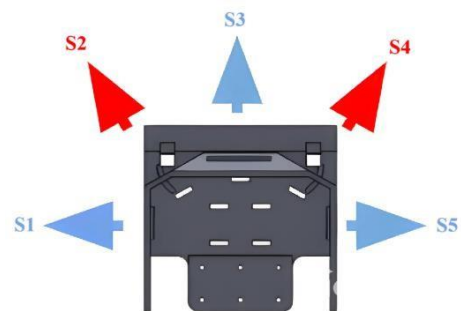


Figure 2. Sensores digitales distribución.

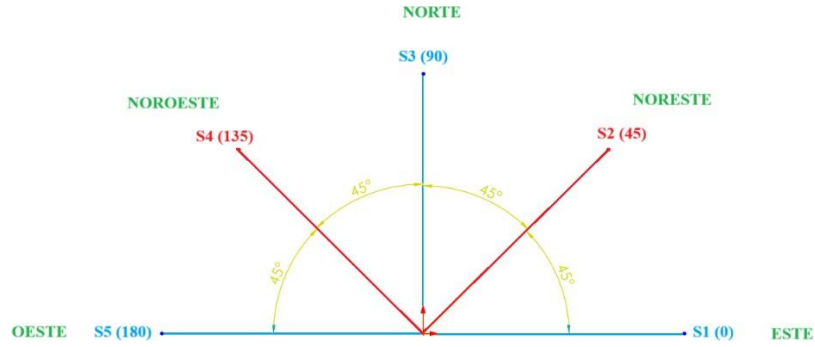


Figure 3. Separación en ángulos de los sensores

## 9. Funciones de membresía

Nuestro sistema cuenta con una entrada y dos salidas (vea Figura 4). Las entradas corresponden a los grados de detección del oponente y los sensores de piso, las salidas están relacionadas con la velocidad de los motores del robot (vea Tabla 2).

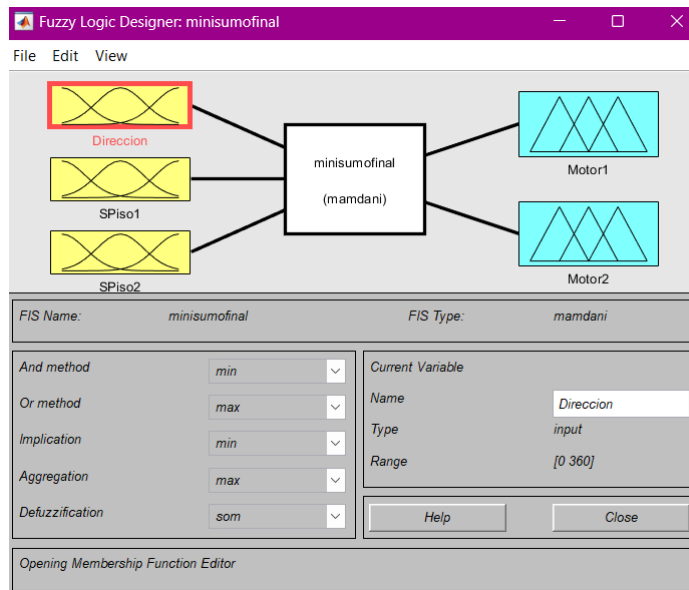


Figure 4. Modelo Fuzzy: entradas y salidas.

Table 1. Variables lingüísticas y funciones de membresía para los grados.

Variable (Entrada)	Grados (0 - 360)	
Variables lingüísticas	Universo del discurso	Funciones de membresía
Este	0 - 45	Triangular
Noreste	35 - 80	Triangular
Norte	70 - 115	Triangular
Noroeste	105 - 150	Triangular
Oeste	140 - 185	Triangular
Ninguno	180 - 360	Trapezoidal

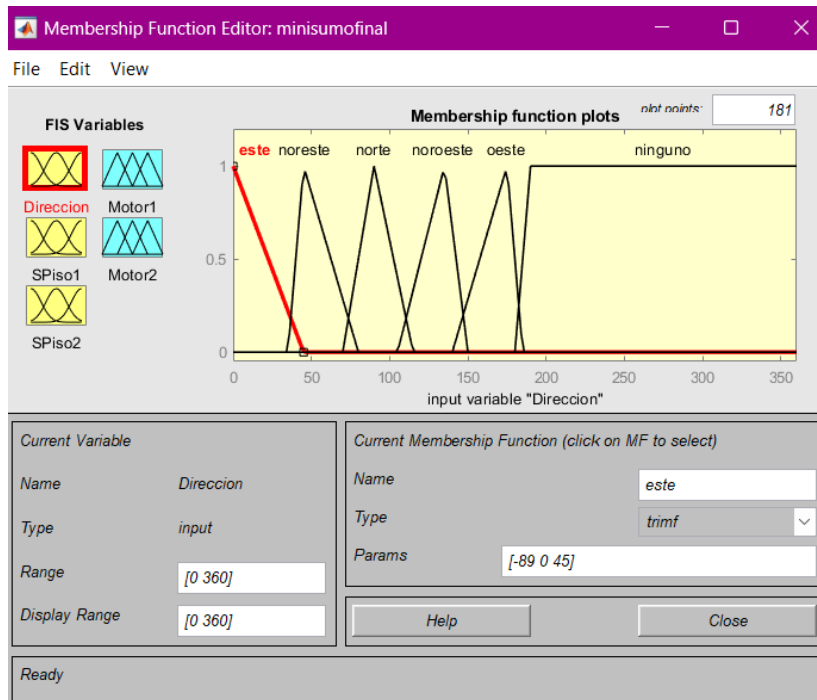


Figure 5. Funciones de membresía dirección.

Table 2. Variables lingüísticas y funciones de membresía para la velocidad.

Variable (Salida)	M1 Velocidad (-100 a 100)	
Variables lingüísticas	Universo del discurso	Funciones de membresía
-VB	-5 a -30	Triangular
-VM	-25 a -65	Triangular
-VA	-60 a -100	Triangular
Quieto	-5 a 5	Triangular
VB	5 a 30	Triangular
VM	25 a 65	Triangular
VA	60 a 100	Triangular

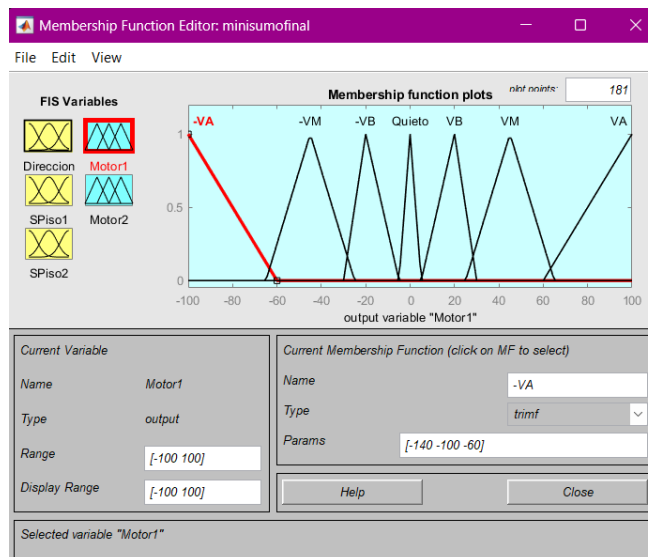


Figure 6. Funciones de membresía motor1 velocidad.

## 10. Reglas

Dirección	SPISO1 (izq)	SPISO2 (der)	M1 Velocidad (Izquierdo)	M2 Velocidad (Derecho)
Este	0	0	VA	-VB
Noreste	0	0	VM	-VM
Norte	0	0	VA	VA
Noroeste	0	0	-VM	VM
Oeste	0	0	-VB	VA
Ninguno	0	0	-VA	VA
Cualquiera	1	0	VA	-VA
Cualquiera	0	1	-VA	VA
Cualquiera	1	1	-VA	-VA

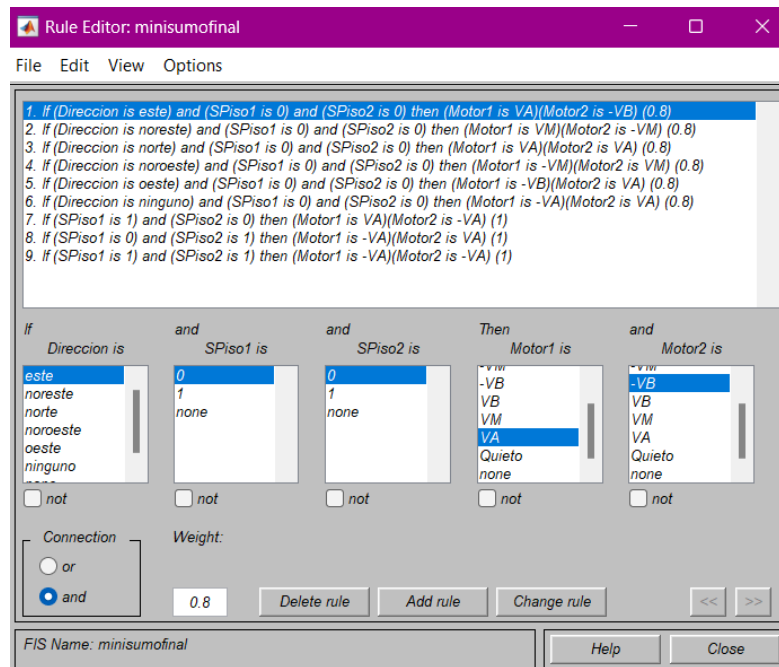


Figure 7. Reglas ingresadas a Matlab.

## 10. Desfuzzificación

El método de desfuzzificación del máximo más pequeño, o mejor conocido como SOM (vea Figura 7), fue el seleccionado para nuestro sistema, ya que proporciona la capacidad de obtener siempre una velocidad regulada por parte de los motores cuando es requerida, sin sacrificar la precisión a la hora de mantener bajas velocidades, apagar el motor o manejar altas velocidades. Este efecto se nota mayormente en la velocidad media, donde se requiere una velocidad no muy alta, pero también se busca que el mini sumo tenga la respuesta más rápida a los estímulos.

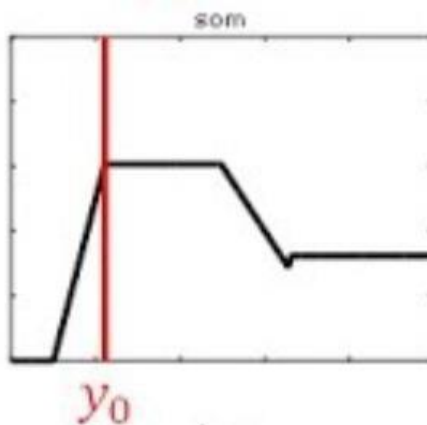


Figure 7. Máximo más pequeño (SOM).

## 11. Sensores de piso

Detectar si hay línea o cambia el color del piso aplica para ambos sensores.

Izquierda= (s\_izq) Derecha= (s\_der)

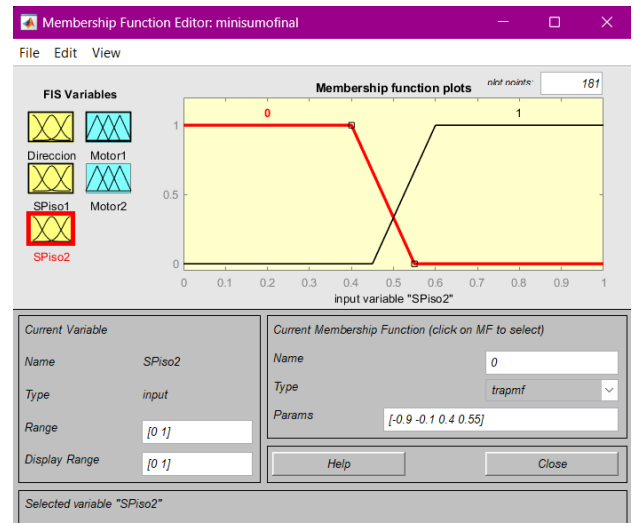
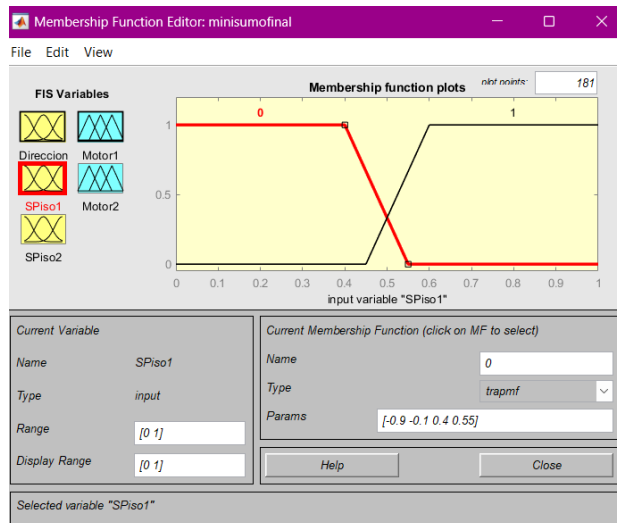
Claro (Esta al borde) Oscuro(No esta al borde) Difusas:

Si = (s\_izq es oscuro) y (s\_der es oscuro) = Avanzar

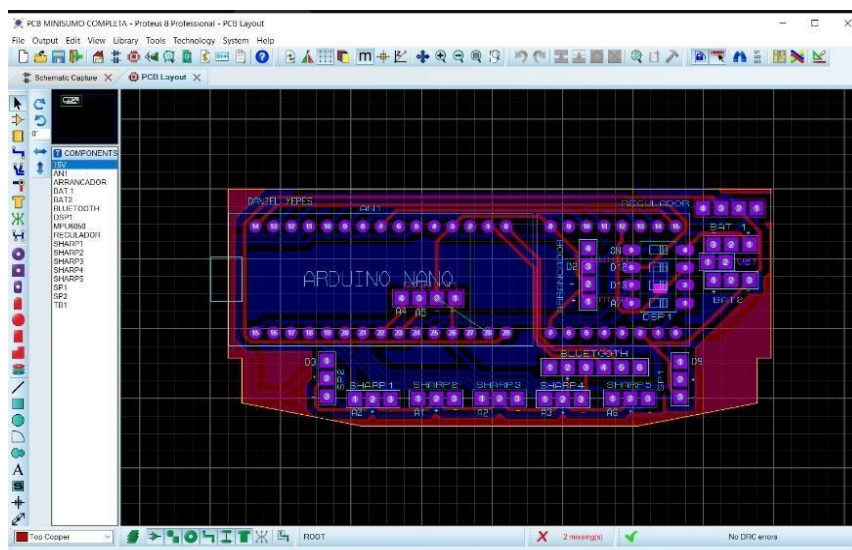
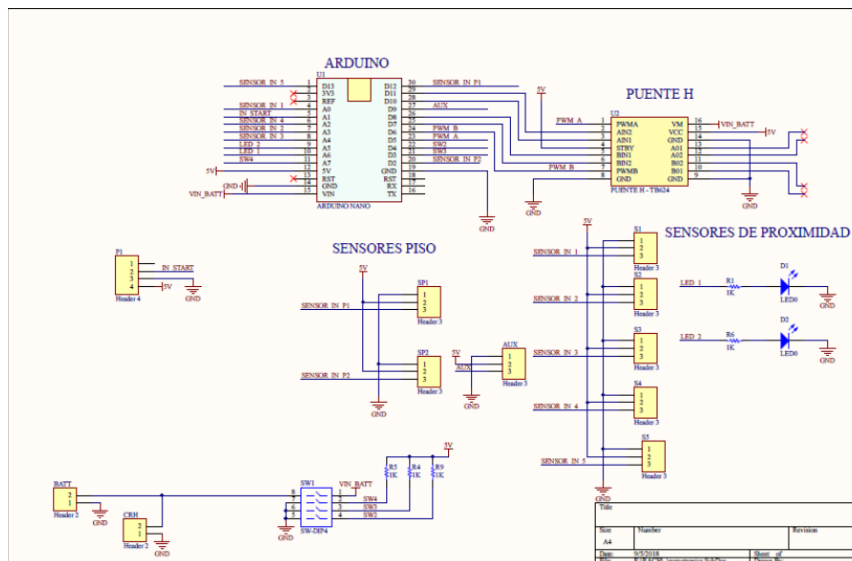
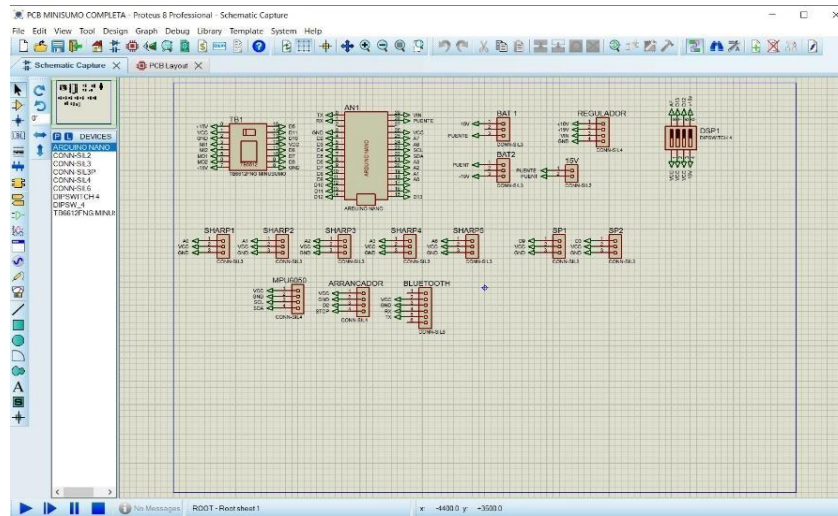
Si = (s\_izq es claro) y (s\_der es oscuro) = Girar ala derecha

Si = (s\_izq es oscuro) y (s\_der es claro) = Girar izquierda

Si = (s\_izq es claro) y (s\_der es claro) = Retroceder

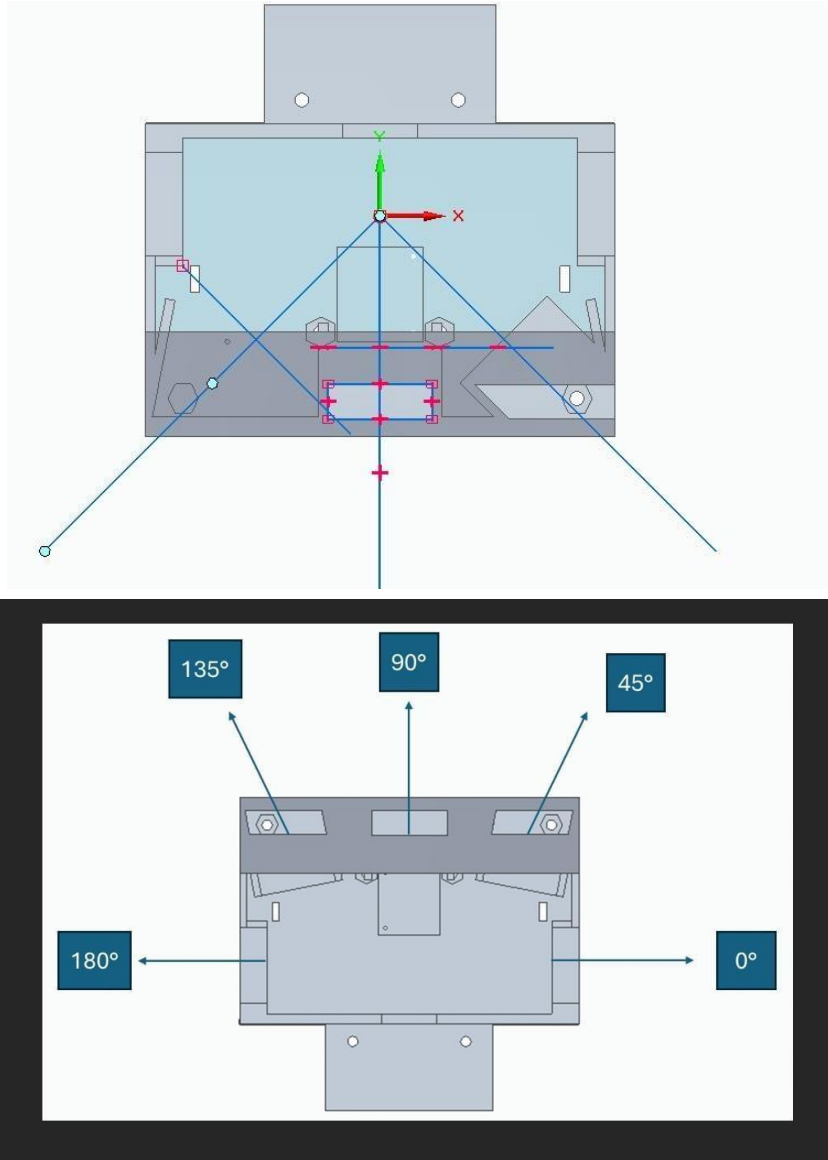


## 12. PCB



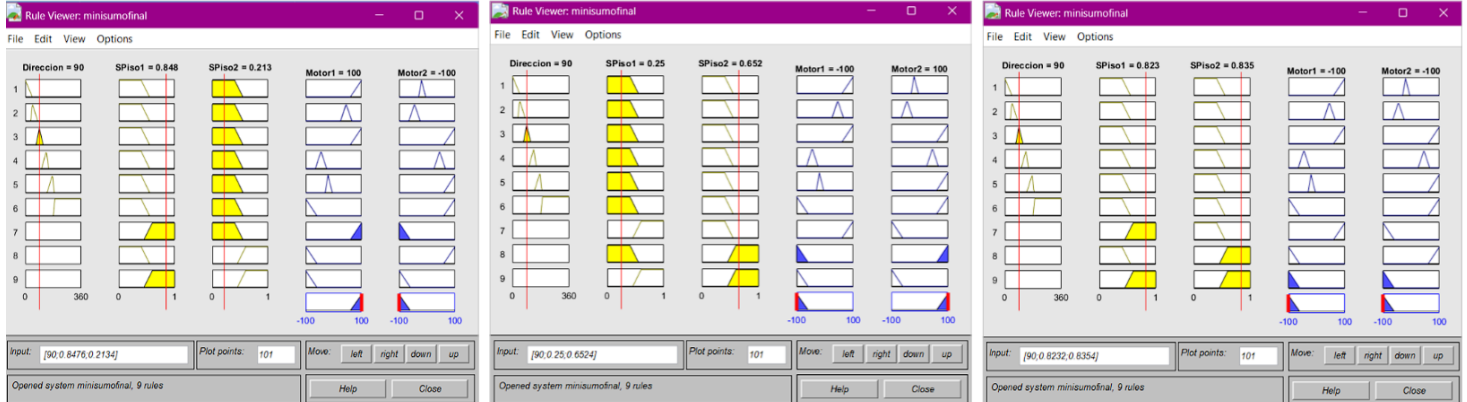


### 13. Diseño.



### 14. Conclusión

Este proyecto de robot mini sumo con control fuzzy presenta un desafío interesante que combina hardware y software avanzados. La implementación de un sistema de control fuzzy proporcionará al robot una ventaja competitiva en términos de toma de decisiones, con una planificación cuidadosa y una ejecución meticulosa, este proyecto tiene el potencial de resultar en un robot mini sumo altamente competitivo.



## 15. Referencias.

- I. Ramírez Campoverde, J. C. (2018) Implementación de algoritmo de lógica difusa utilizando el microcontrolador Atmega 32u4 como estrategias de lucha en la categoría minisumo (Trabajo de titulación)<sup>12</sup>. Universidad Cata de Santiago de Guayaquil, Facultad de Educación Técnica para el Desarrollo, Carrera de Ingeniería en Telecomunicaciones.
- II. Erdem, H. (2007). A Practical Fuzzy Logic Controller for Sumo Robot Competition. En A. Ghosh, R. K. De, & S. K. Pal (Eds.), Pattern Recognition and Machine Intelligence (pp. 217-225). Springer-Verlag Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-77046-6\\_27](https://doi.org/10.1007/978-3-540-77046-6_27).
- III. Alfaro Cruz, M. A., Gutiérrez Velázquez, L. D. E., & Gorrostieta, E. V. (2013). Sistema de Lógica Difusa Para el Reconocimiento y Evasión de Obstáculos. En 12º Congreso Nacional de Mecatrónica (pp. 173-177). León, Guanajuato: Asociación Mexicana de Mecatrónica A.C.
- IV. Erdem, H. (2011). Application of Neuro-Fuzzy Controller for Sumo Robot control. Expert Systems with Applications, 38(8), 9752-9760.  
1. <https://doi.org/10.1016/j.eswa.2011.02.024>.