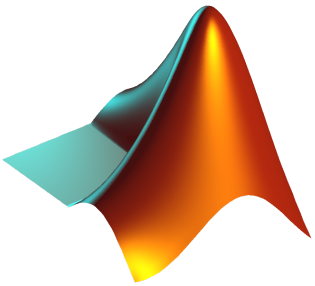
Universidad de Alcalá

Sistemas de Control Inteligente

Tema: Práctica 2. Control Borroso



Grupo A2_P09

Integrantes

Ana Cortés Cercadillo

Carlos Javier Hellín Asensio

Grado de Ingeniería Informática

Curso: 2021-2022

Índice

Parte 1. Diseño de un control borroso de posición para un robot móvil.

Para empezar creando el controlador borroso se puede acceder a las herramientas de “Fuzzy Logic Designer” mediante la interfaz gráfica de MATLAB en “Apps -> Fuzzy Logic Designer”.

Figura 1. Acceder desde el menú al Fuzzy Logic Designer

O también ejecutando el comando fuzzy desde la ventana de comandos.

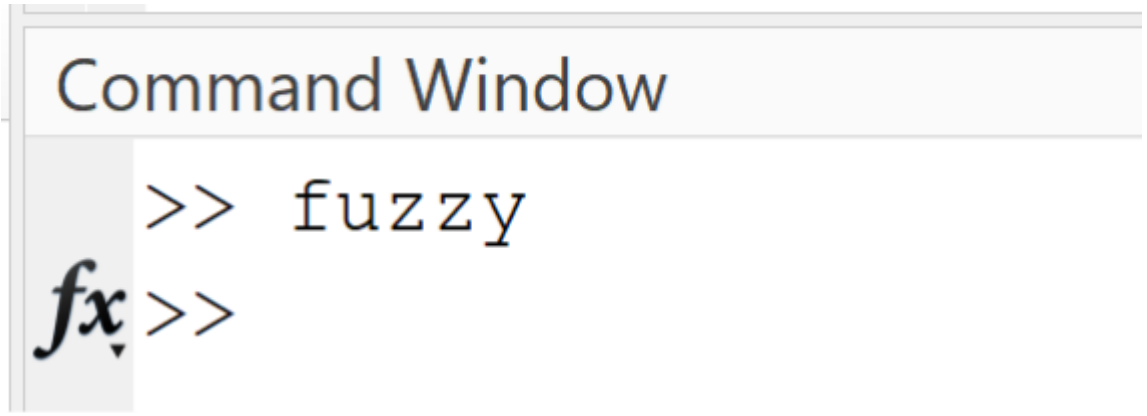


Figura 2. Ejecutando el comando fuzzy.

Una vez hecho esto, se tiene la siguiente interfaz:

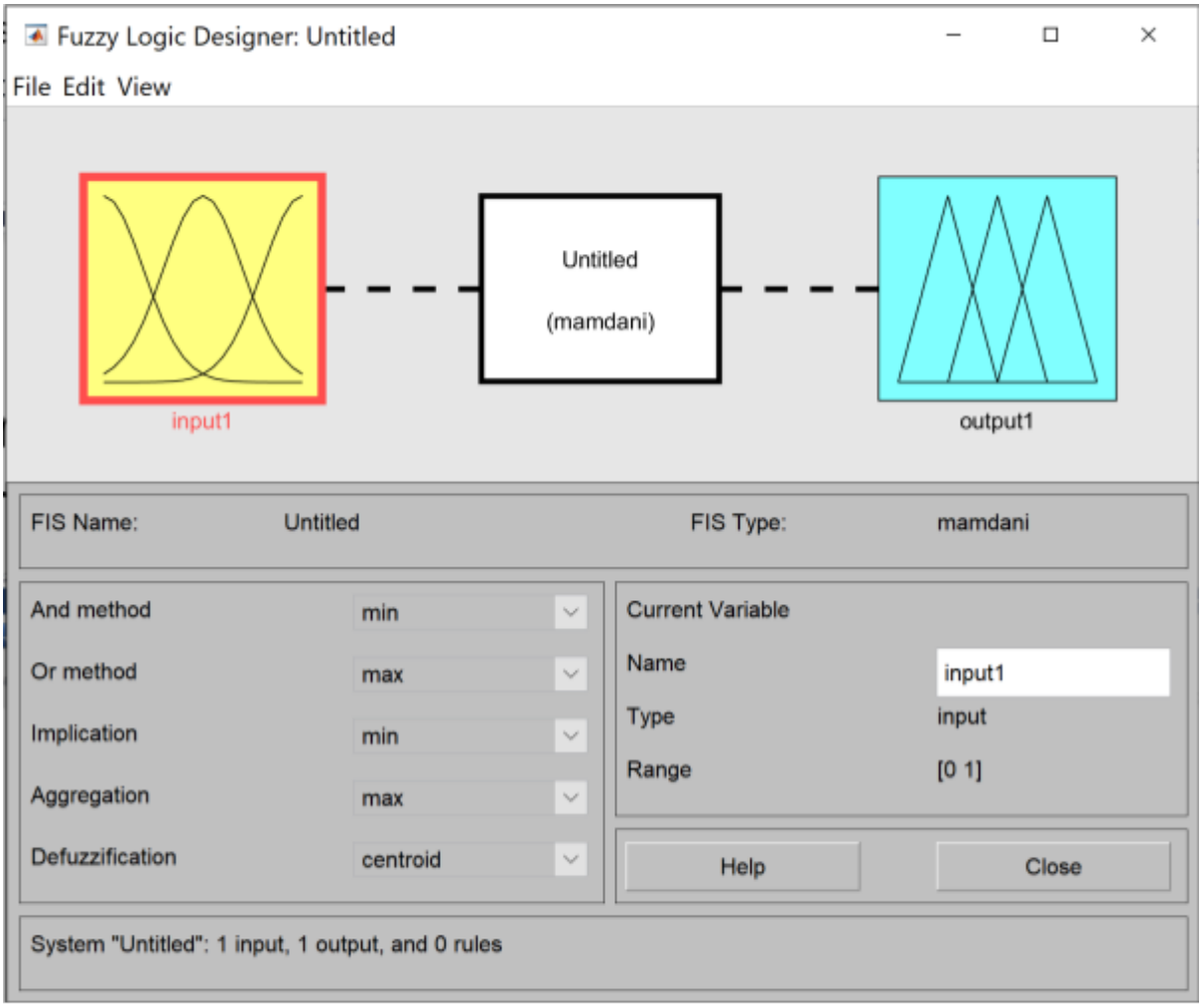


Figura 3. La interfaz de Fuzzy Logic Designer.

Se crean las variables (tanto de entrada como de salida) desde el menú en “Edit -> Add Variable.. -> Input/Output”.

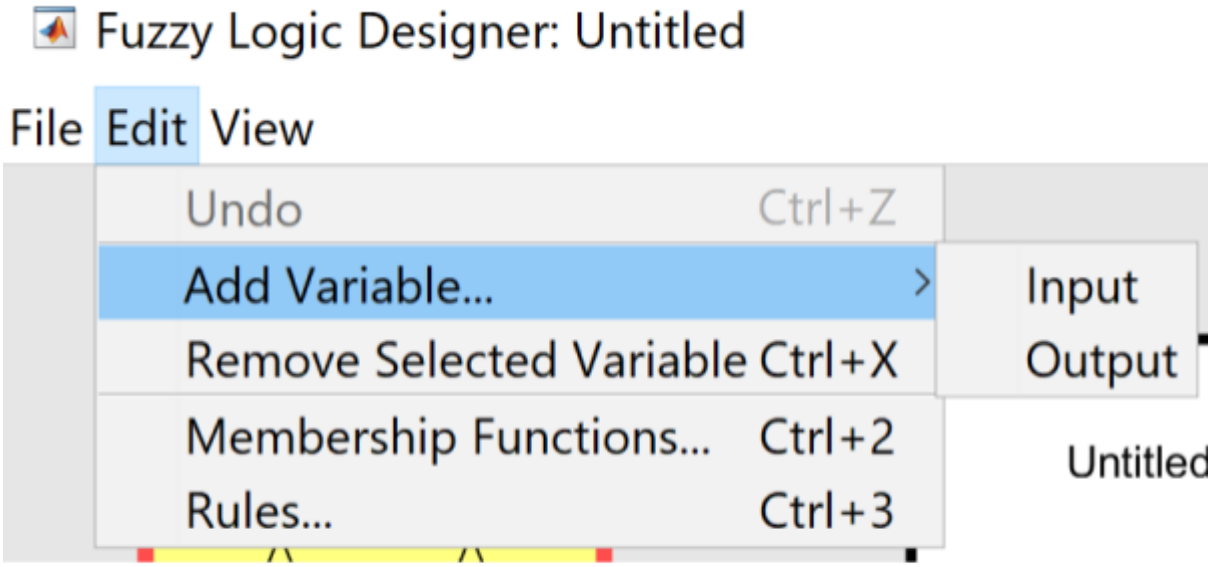


Figura 4. Añadir una variable (Entrada o salida).

De esta forma se diseña el controlador borroso como se propone en el enunciado dando nombres a las variables desde la caja de texto “Name”.

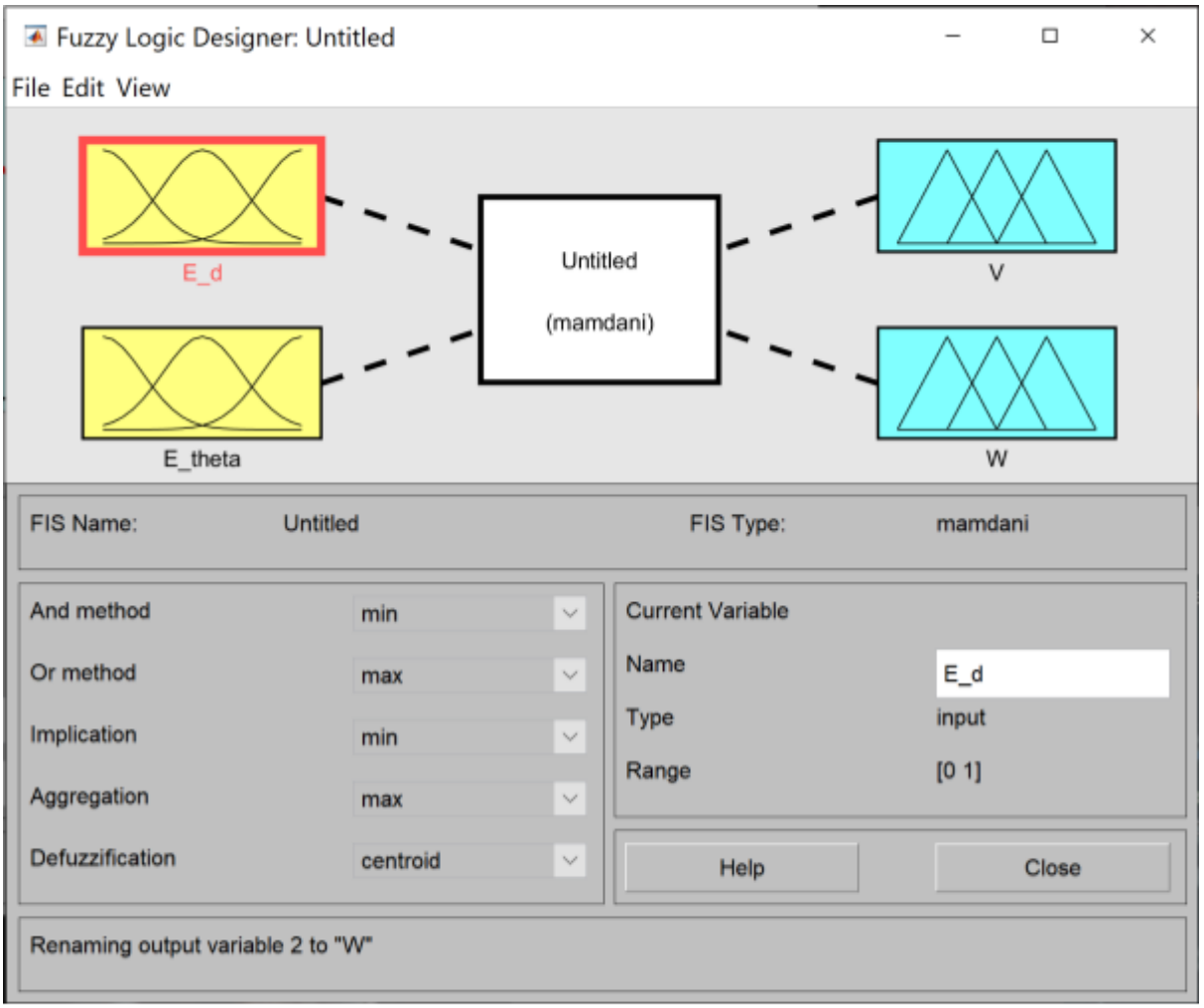
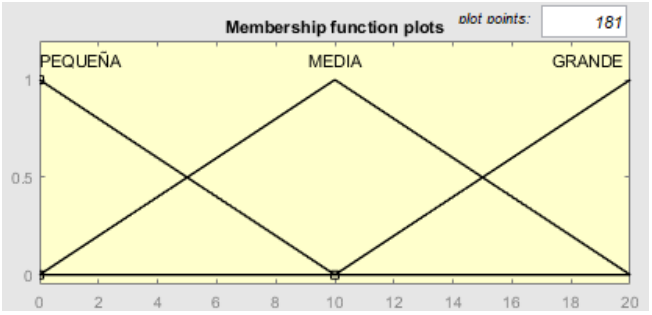
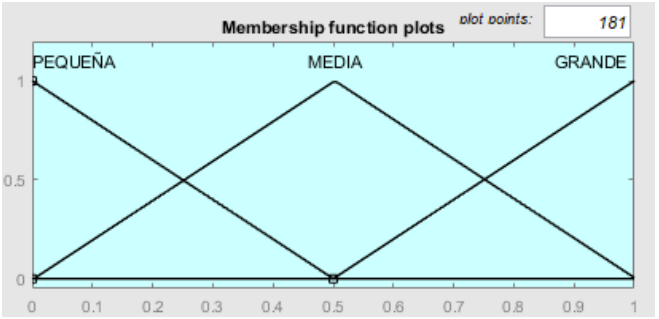
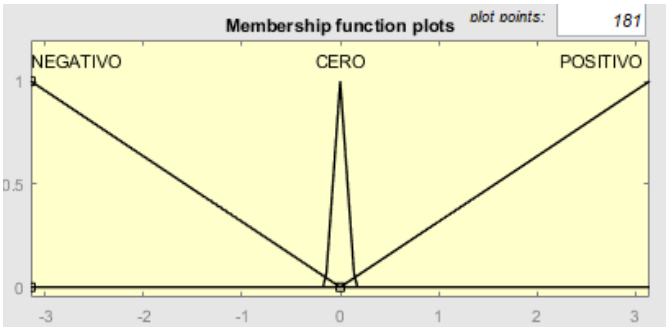
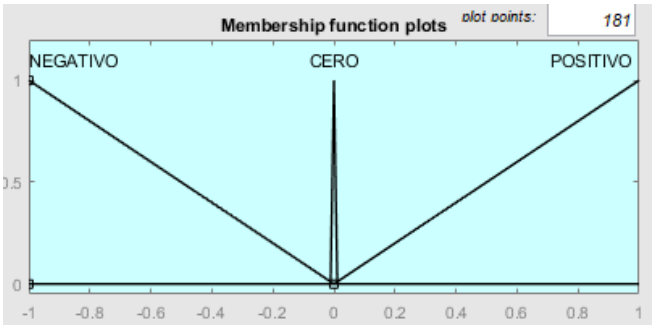


Figura 5. Se añaden las variables y se les da nombres.

Se han ajustado las variables y dado nombres a los conjuntos borrosos. Se ha modificado la velocidad lineal para un rango de 0 a 1 en vez de al rango propuesto en el enunciado (de 0 a 2), y el error de theta de tal forma que se consiga corregir la trayectoria con una mayor precisión:

E_d	**V**
	
E_theta	**W**
	

Creamos las gráficas de las funciones de cada variable con los siguientes datos:

- **E_d:** el rango de la variable es [0, 20] y se compone de tres funciones:
  - PEQUEÑA: el pico lo alcanza en el valor x=0 y desciende hasta x=10.
  - MEDIA: de x=0 hasta x=10 la función sube, alcanza su pico y luego baja hasta x=20.
  - GRANDE: en x=10 empieza a subir hasta llegar a su máximo en x=20.
- **E_theta:** el rango de esta variable es [-3.142, 3.142] y tiene tres funciones:
  - NEGATIVO: en x=-3.142 alcanza su pico y desciende hasta x=0.
  - CERO: en x=-0.15 empieza a subir hasta llegar a su pico en x=0 y desciende hasta x=0.15.
  - POSITIVO: a partir de x=0 comienza a subir hasta su pico en x=3.142.
- **V:** el rango es [0, 1] y se compone de tres funciones:
  - PEQUEÑA: su pico más alto es en x=0 hasta bajar a x=0.5.
  - MEDIA: empieza a subir en x=0 hasta llegar a x=0.5 y luego baja hasta x=1.
  - GRANDE: en x=0.5 empieza a subir hasta x=1.
- **W:** el rango es [-1, 1] y tiene tres funciones:
  - NEGATIVO: el pico lo tiene es x=-1 y va bajando hasta x=0.
  - CERO: es una delta con valor x=0.
  - POSITIVO: En x=0 empieza a subir hasta llegar a su pico en x=1.

Posteriormente se añaden las reglas mediante el “Rule Editor” que está disponible desde el menú “Edit -> Rules…”.

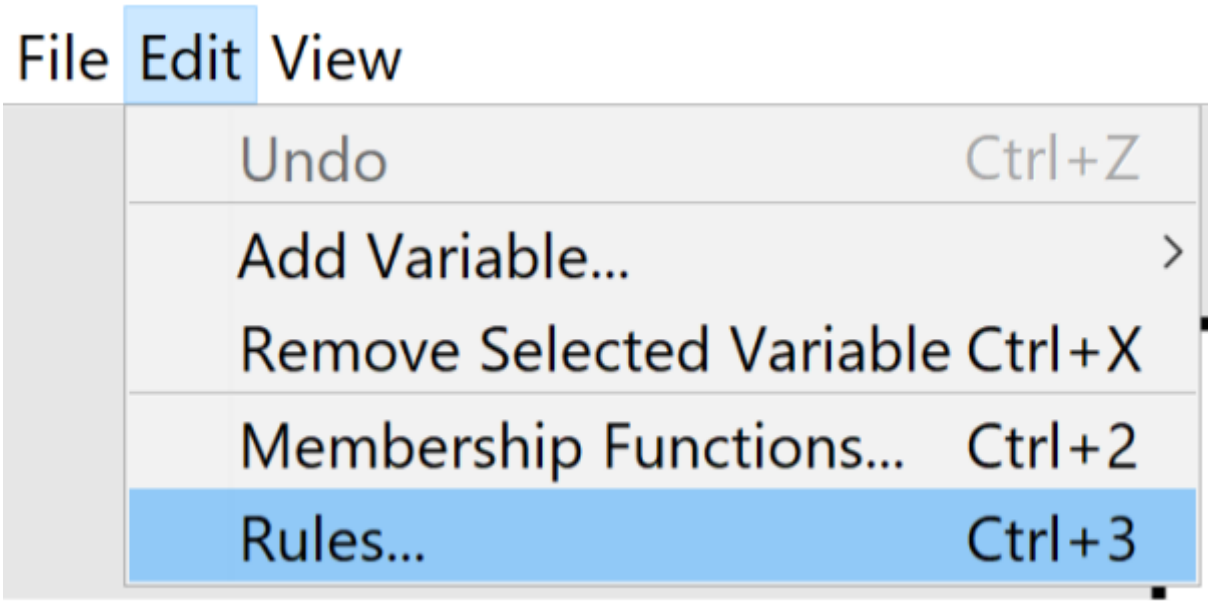


Figura 6. Añadir reglas al Controlador Borroso.

Otra forma de abrir la ventana de las reglas sería haciendo doble click sobre la figura que representa al controlador borroso (cuadrado blanco con bordes negros que pone “mamdani”) en la ventana de la Figura 5.

Se han definido las siguientes reglas en el controlador, de forma que haya una separación en función de su ámbito, ya sea por distancia o ángulo como se puede ver en la Figura 7.

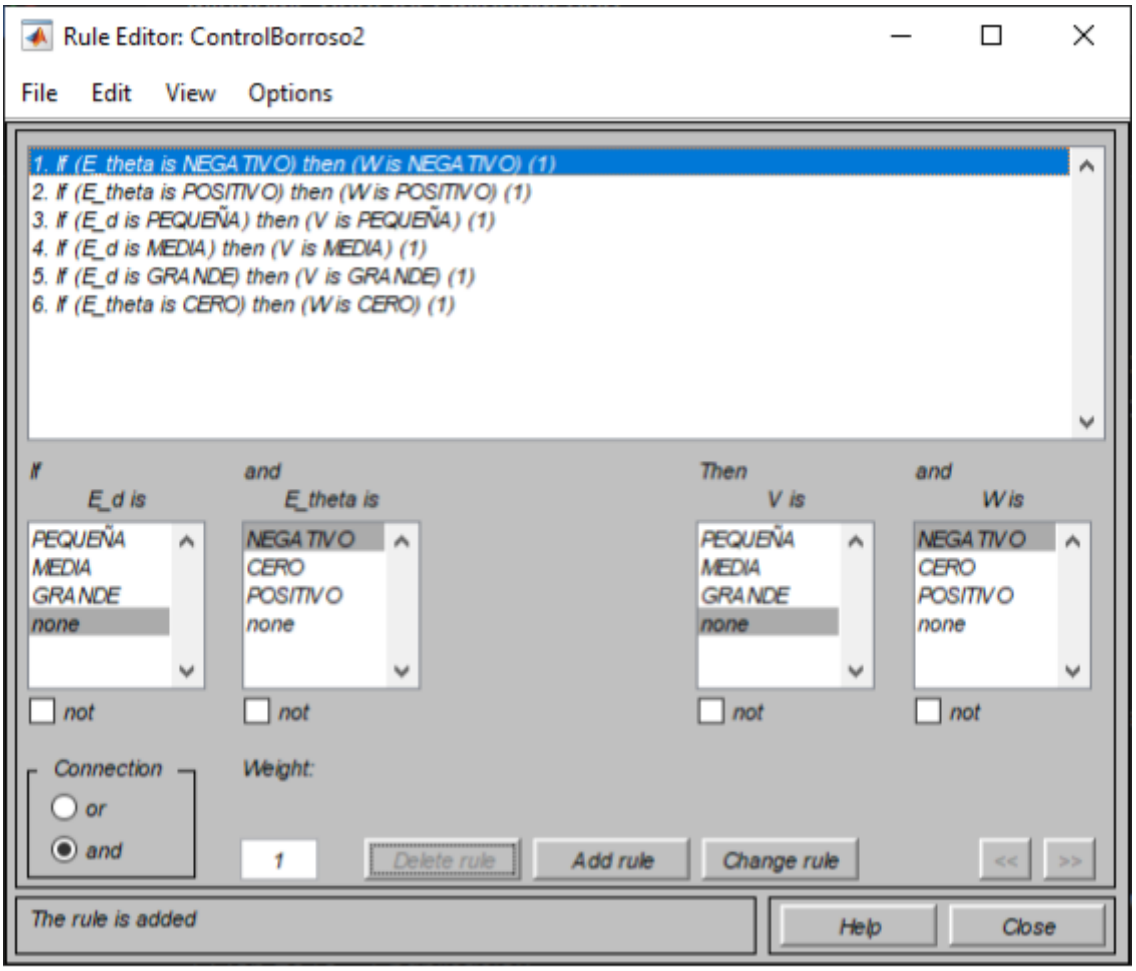


Figura 7. Las reglas definidas del controlador.

Se guarda el controlador borroso a un fichero .fis y se diseña el siguiente esquema con un bloque que contiene el controlador borroso con las dos entradas y salidas:

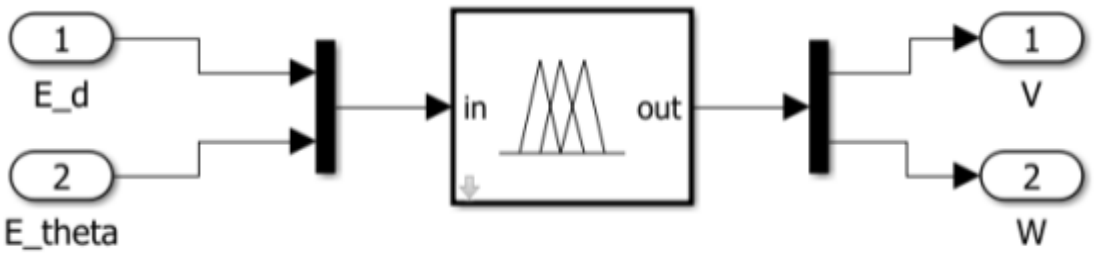


Figura 8. Se crea el bloque controlador borroso.

Y se sustituye el bloque Control por el del Controlador Borroso dejando así el diagrama:

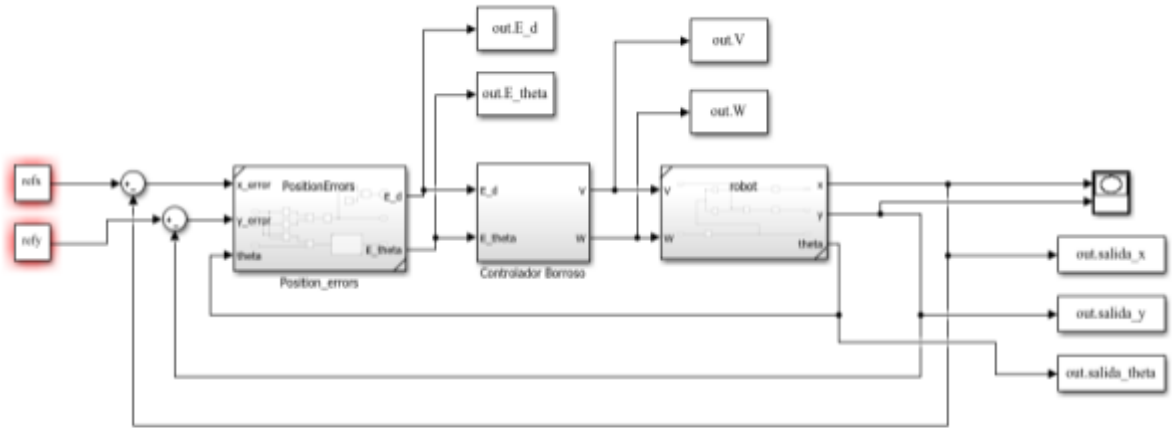


Figura 9. Diagrama final con el Controlador Borroso.

Con este script se prueba su funcionamiento cambiando los valores de refx y refy.

```
%Tiempo de muestreo
Ts=100e-3
% Referencia x-y de posicion
refx=10*rand
refy=10*rand
% Ejecutar Simulacion
sim('PositionControl.slx')
% Mostrar
x=salida_x.signals.values;
y=salida_y.signals.values;
figure;
plot(x,y);
title("Trayectoria seguida por el robot");
xlabel("x");
ylabel("y");
grid on;
hold on;
```

Figura 10. Script utilizado para comprobar su funcionamiento.

El robot se inicia en la posición 0,0 y usando rand en una primera ejecución se obtiene el refx = 5.4688 y refy = 9.575. La gráfica resultante es la siguiente que muestra que el robot consigue ir al punto dado avanzando en línea recta:

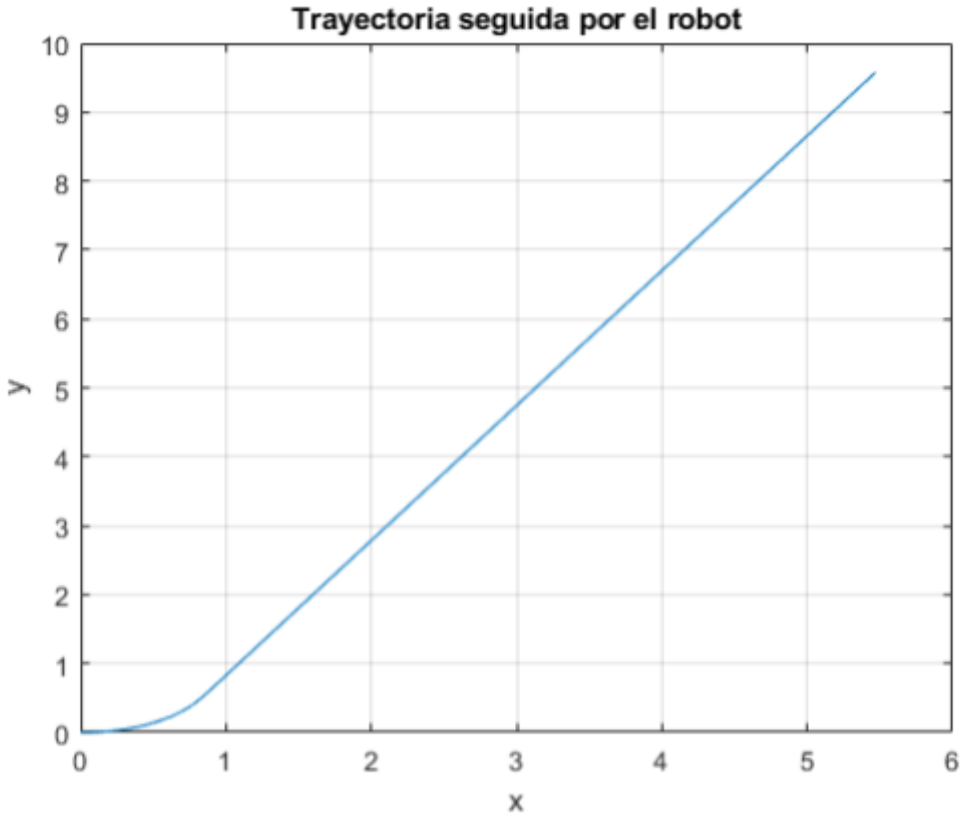


Figura 11. El robot avanza en línea recta hasta su destino.

Cuando refx = -5 y refy = -5 (sin usar el rand, se han metido los datos a mano) empieza corrigiendo la trayectoria y llega a su destino en línea recta.

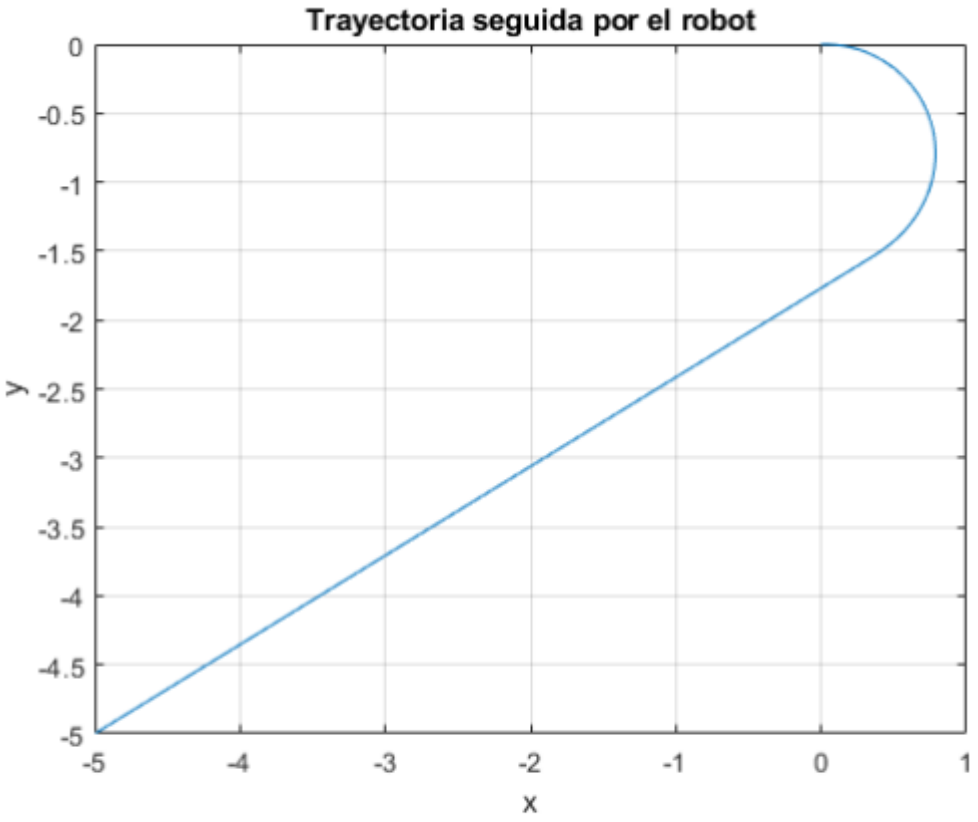


Figura 12. El robot consigue corregir la trayectoria.

Ahora se sustituye las referencias de posición por el generador de trayectorias manteniendo el Controlador Borroso con algunos cambios necesarios para que el robot consiga seguir la trayectoria.

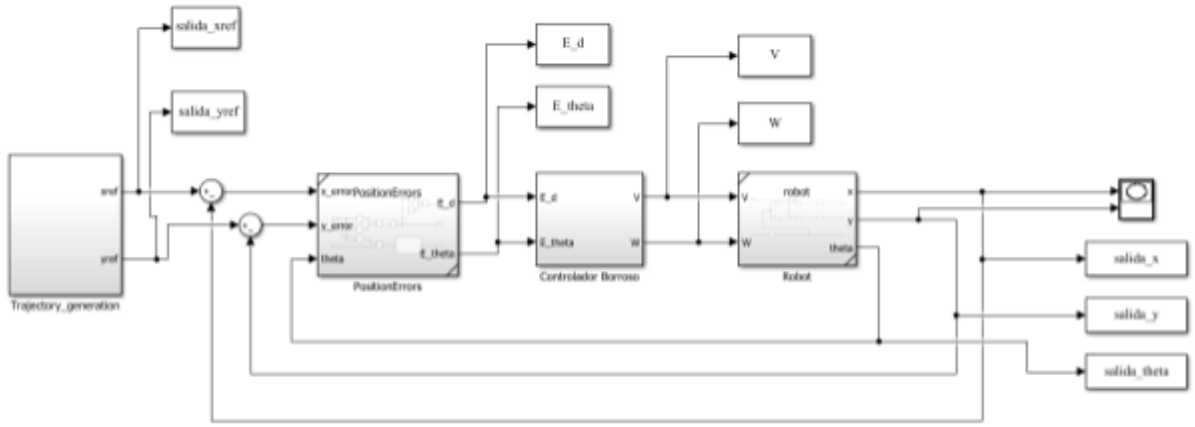


Figura 13. Diagrama anterior con el generador de trayectorias.

El script que se va a utilizar es el siguiente para generar una gráfica que permita comparar la trayectoria generada con la del robot.

```
clear all;
close all;

% Tiempo de muestreo
Ts = 100e-3;

% Parámetros iniciales de la trayectoria
x_0 = 0;
y_0 = 0;
th_0 = 0;

% Ejecutar Simulación
sim('TrajectoryControl.slx');

xref = salida_xref.signals.values';
yref = salida_yref.signals.values';
x = salida_x.signals.values';
y = salida_y.signals.values';

figure(1);
hold on;
generada = plot(xref, yref);
robot = plot(x, y);
hold off;
grid on;
legend('Trayectoria generada', 'Trayectoria robot');
title('Comparación de las trayectorias');
xlabel('Eje X');
ylabel('Eje Y');
```

Figura 14. Script utilizado para comparar la trayectoria generada y la del robot.

Antes de ejecutar el script, se ha tenido que hacer ajustar el rango de E_d para que sea entre 0 y 10:

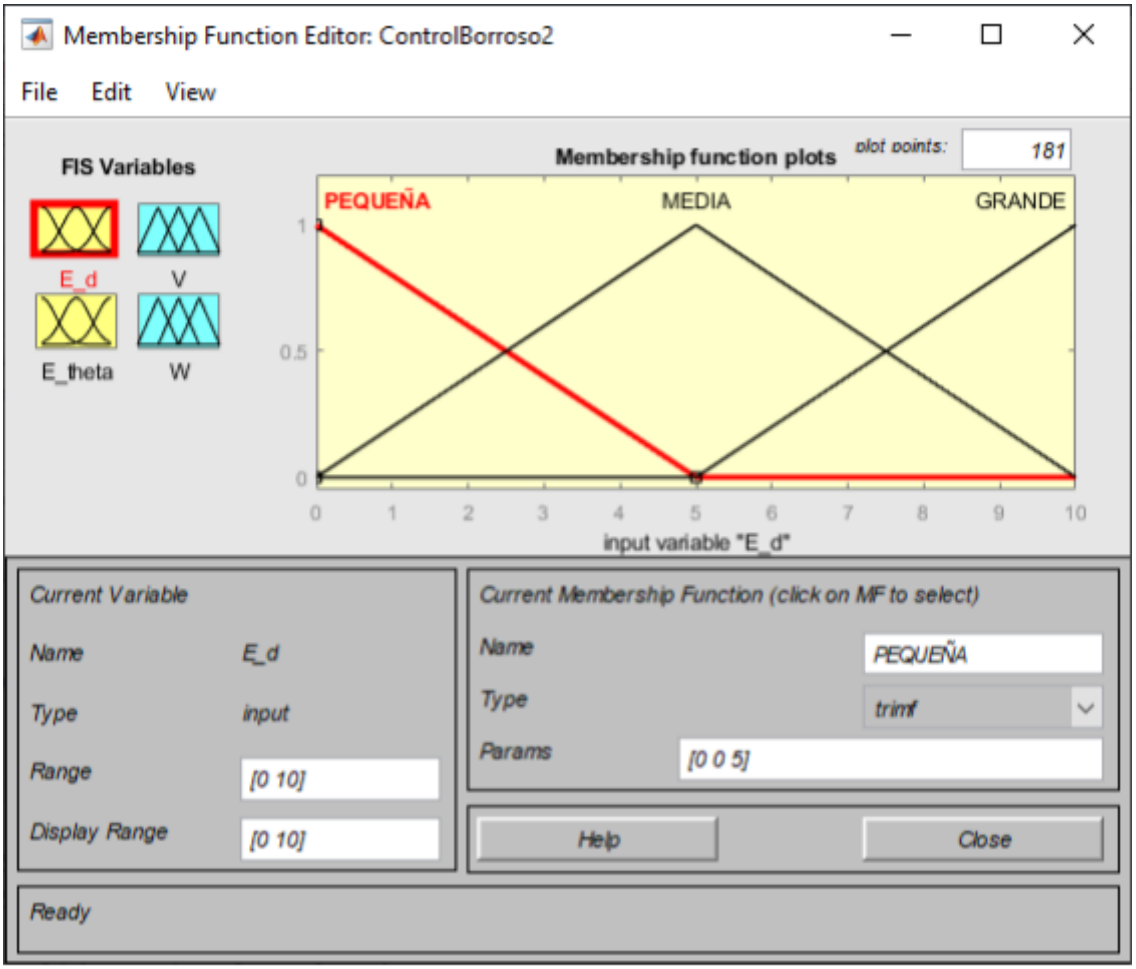


Figura 15. Se modifica el rango de E_d.

Una vez hecho esto, ya se ejecuta el script y se puede observar que el robot consigue realizar la trayectoria con bastante precisión:

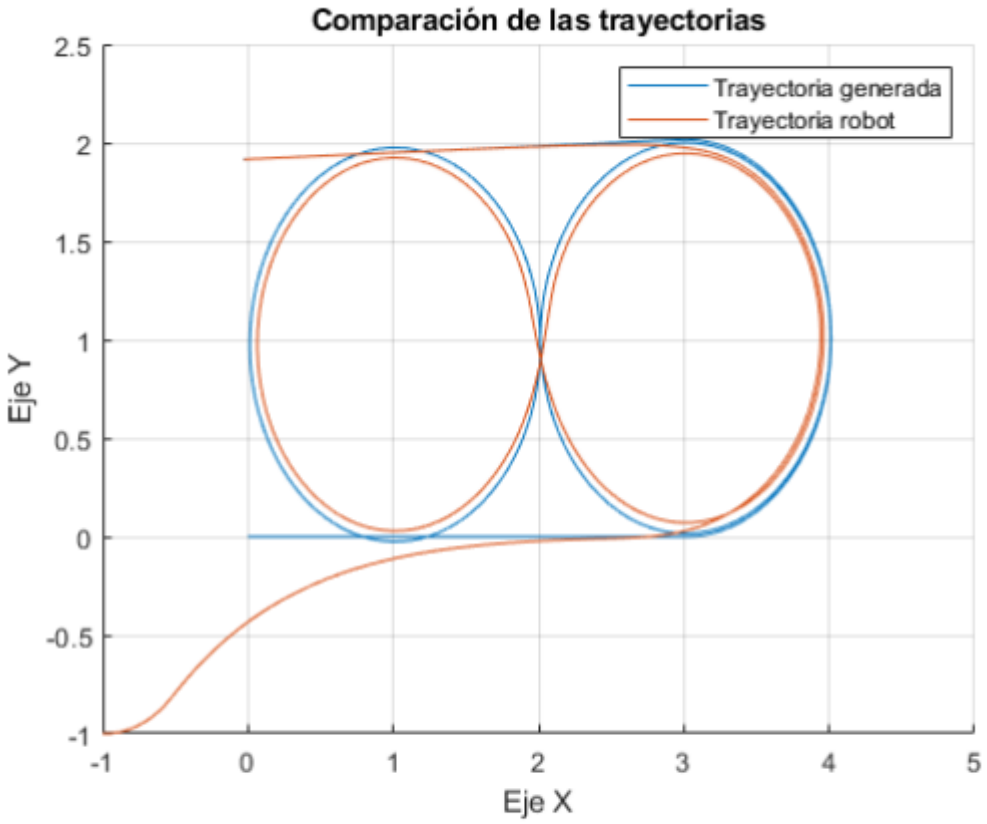


Figura 16. Gráfica que compara las dos trayectorias.

Parte 2. Diseño de control borroso de posición con evitación de obstáculos

Usando el controlador de la parte 1, se diseña el siguiente controlador borroso, con sus dos nuevas variables de entrada:

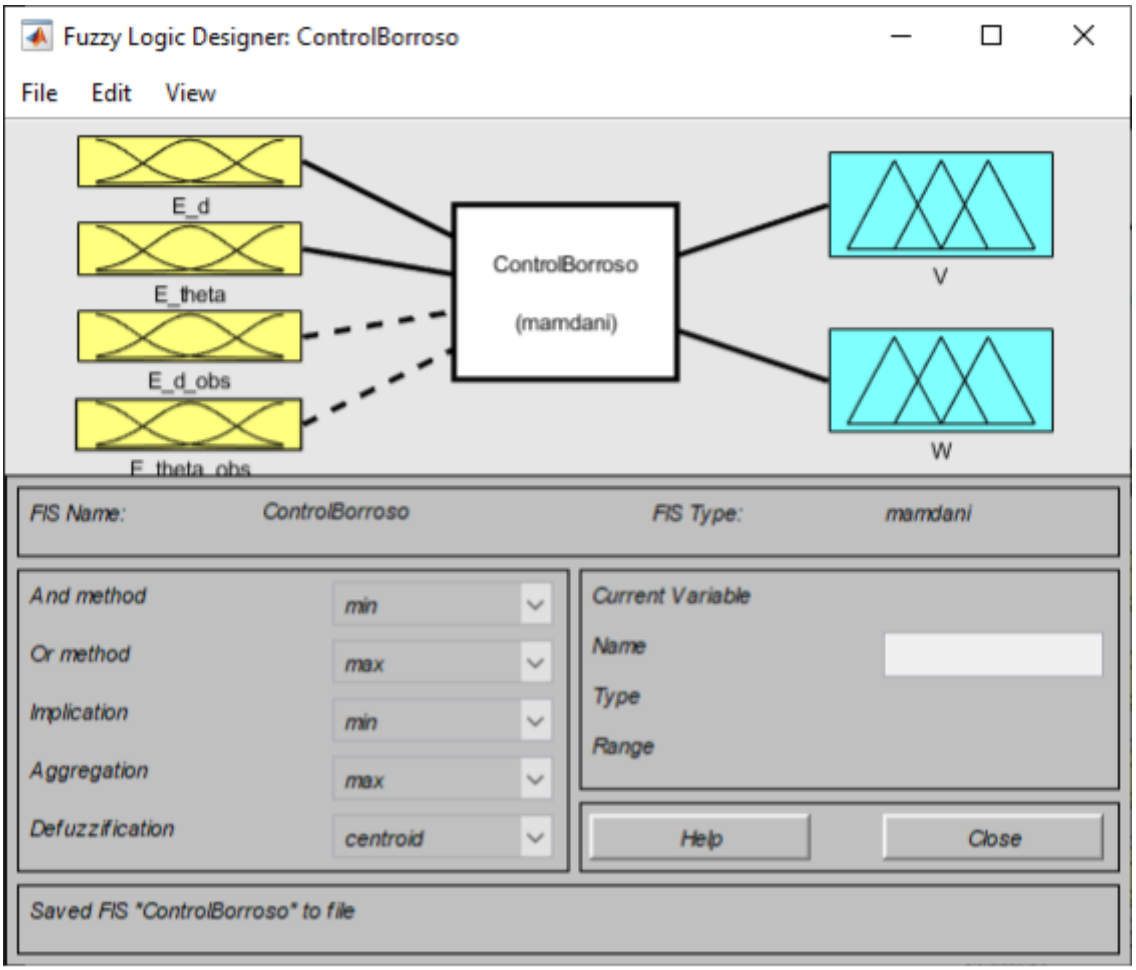
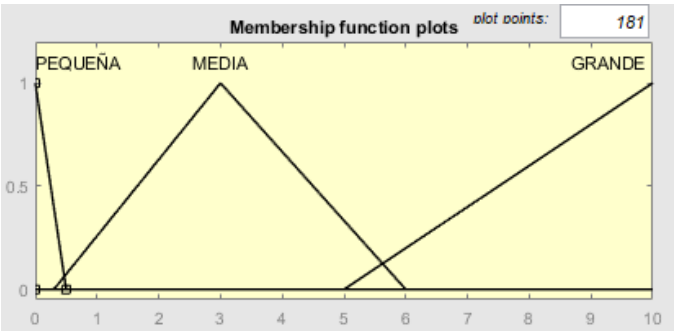
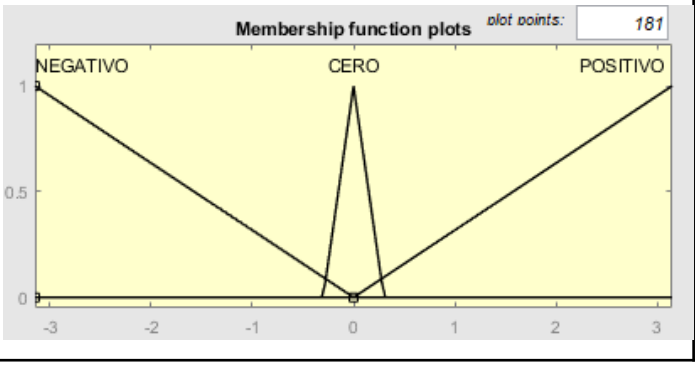


Figura 17. Se añaden dos variables nuevas y se les da un nombre.

Se diseñan las funciones de pertenencia del error de distancia y el ángulo del obstáculo.

E_d_obs	E_theta_obs
	

Creamos las nuevas gráficas de las funciones de cada variable con los siguientes datos:

- **E_d_obs:** el rango de la variable es [0, 10] y se compone de tres funciones:
  - PEQUEÑA: el pico lo alcanza en el valor x=0 y desciende hasta x=0.5.
  - MEDIA: de x=0.3 hasta x=3 la función sube, alcanza su pico y luego baja hasta x=6.
  - GRANDE: en x=5 empieza a subir hasta llegar a su máximo en x=10.
- **E_theta_obs:** el rango de esta variable es [-3.142, 3.142] y tiene tres funciones:
  - NEGATIVO: en x=-3.142 alcanza su pico y desciende hasta x=0.
  - CERO: en x=-0.3 empieza a subir hasta llegar a su pico en x=0 y desciende hasta x=0.3.
  - POSITIVO: a partir de x=0 comienza a subir hasta su pico en x=3.142.

Estas funciones de trayectoria han sido establecidas así para tener una corrección lo más cerca posible del obstáculo. De esta forma se consigue que la trayectoria final se vea lo menos afectada por el obstáculo que se pueda encontrar.

Se añaden las siguientes reglas al controlador:

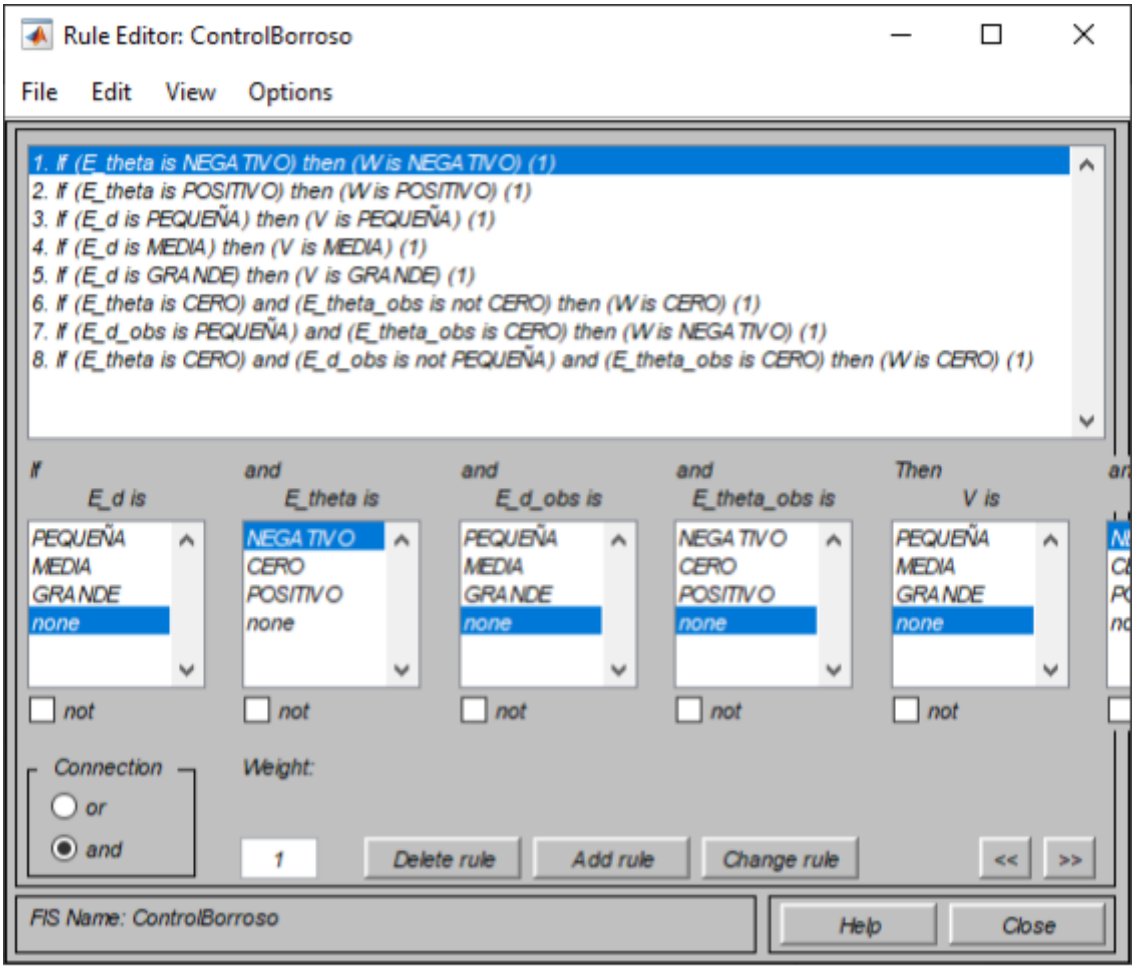


Figura 18. Las nuevas reglas para corregir la trayectoria respecto al obstáculo.

Se puede observar que se han añadido dos reglas más y se ha modificado una de ellas para la corrección de la trayectoria respecto al obstáculo:

- La regla número 6 quiere decir que cuando el robot no vaya en dirección al obstáculo que continúe su trayectoria.
- La regla número 7 implica que cuando el robot esté al lado del obstáculo y vaya directo a él, se corrige la trayectoria con W negativo, es decir, esto haría que el robot gire a la izquierda.
- La regla número 8 ayuda a que aunque el robot vaya en dirección al obstáculo, no empiece a corregir la trayectoria hasta que esté demasiado cerca del obstáculo.

Como antes, se vuelve a crear el bloque de control borroso con las dos nuevas entradas:

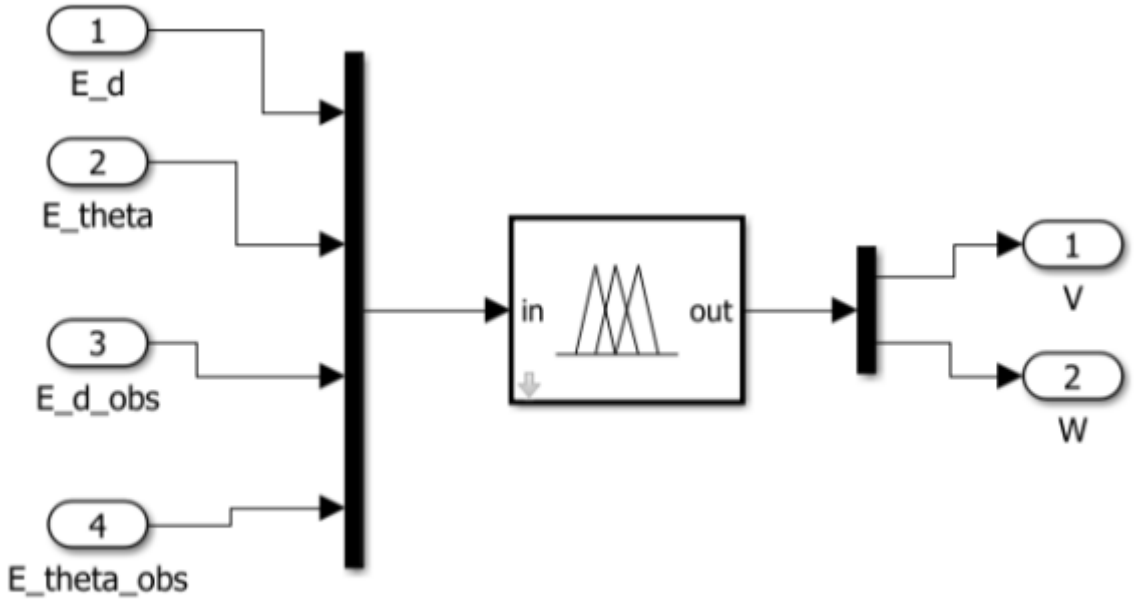


Figura 19. El bloque controlador borroso con las dos nuevas variables de entrada.

Y se modifica el diagrama de bloques con el nuevo Controlador Borroso y la posición obsx y obsy.

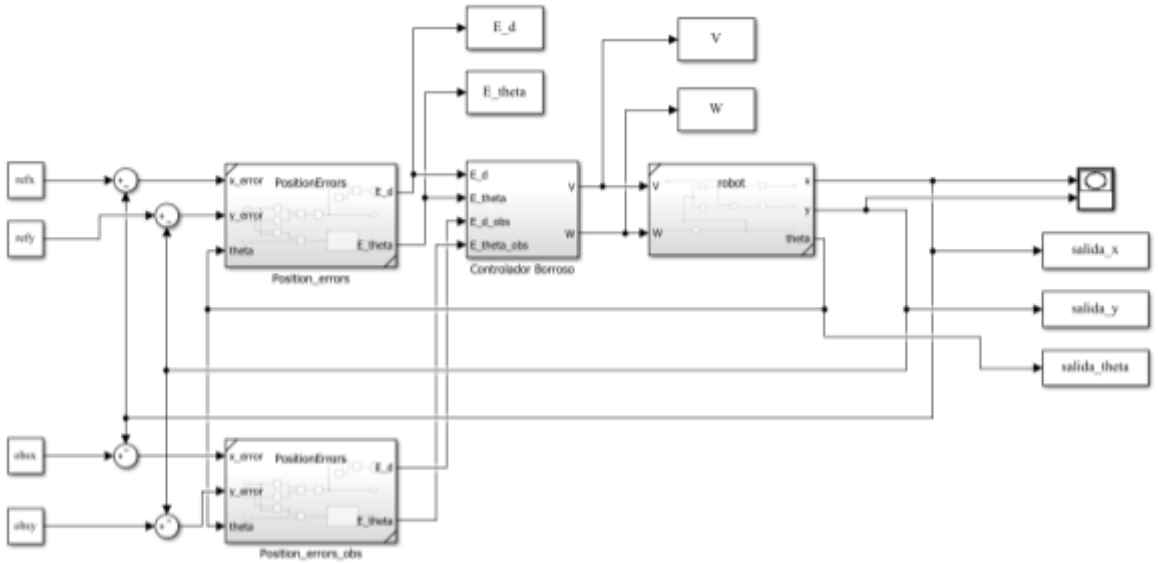


Figura 20. Diagrama final con el Controlador Borroso y la posición del obstáculo.

El script que se va a utilizar es el siguiente, con la gráfica que muestra tanto la trayectoria del robot como del punto dónde se encuentra el obstáculo.

```
%Tiempo de muestreo
Ts=100e-3

% Referencia x-y de posicion
refx=5;
refy=5;
obsx=2.5;
obsy=2.5;

% Ejecutar Simulacion
sim('EvitarObstaculo.slx')

% Mostrar
x=salida_x.signals.values;
y=salida_y.signals.values;

figure;
hold on;
trayectoria = plot(x, y);
obstaculo = plot(obsx, obsy, 'x');
hold off;
grid on;
legend('Trayectoria', 'Obstaculo');
title('Trayectoria del robot con obstaculo');
xlabel('Eje X');
ylabel('Eje Y');
```

Figura 21. Script utilizado para verificar su correcto funcionamiento.

Cuando las variables son: refx = 5, refy = 5, obsx = 2.5 y obsy = 2.5 se obtiene la siguiente gráfica donde se puede observar que no tiene ningún problema en seguir la trayectoria:

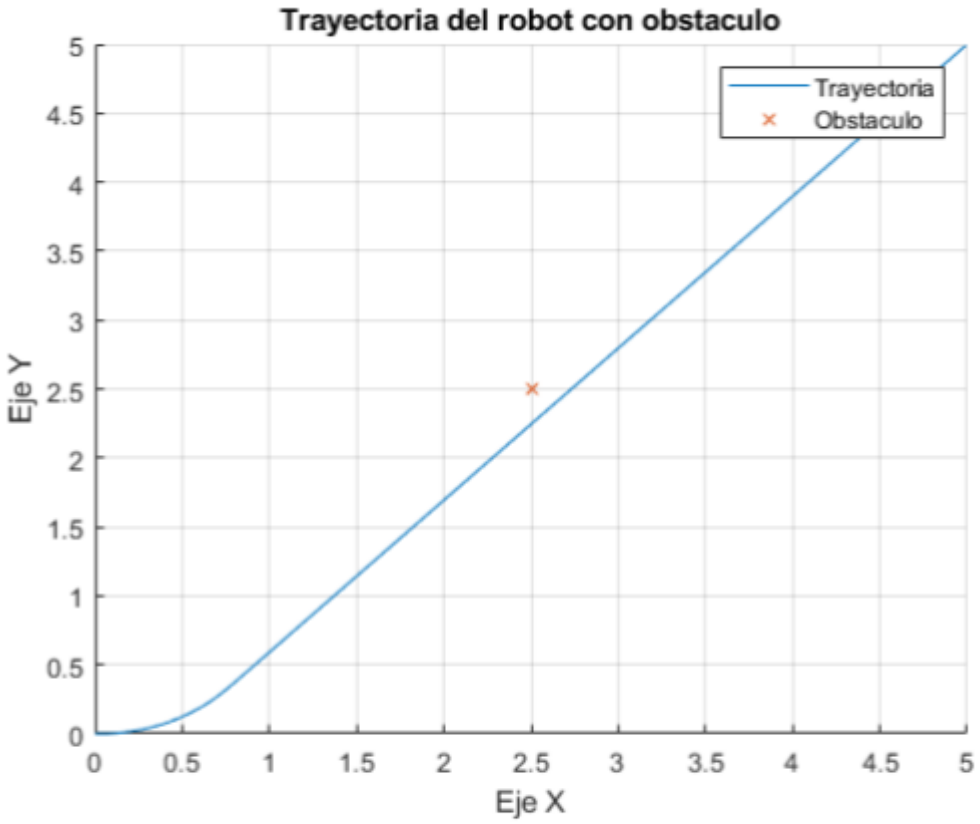


Figura 22. El robot continúa su trayectoria sin problema.

Cuando refx = 9, refy = 0, obsx = 4 y obsy=0, se puede observar como el robot consigue esquivar el obstáculo:

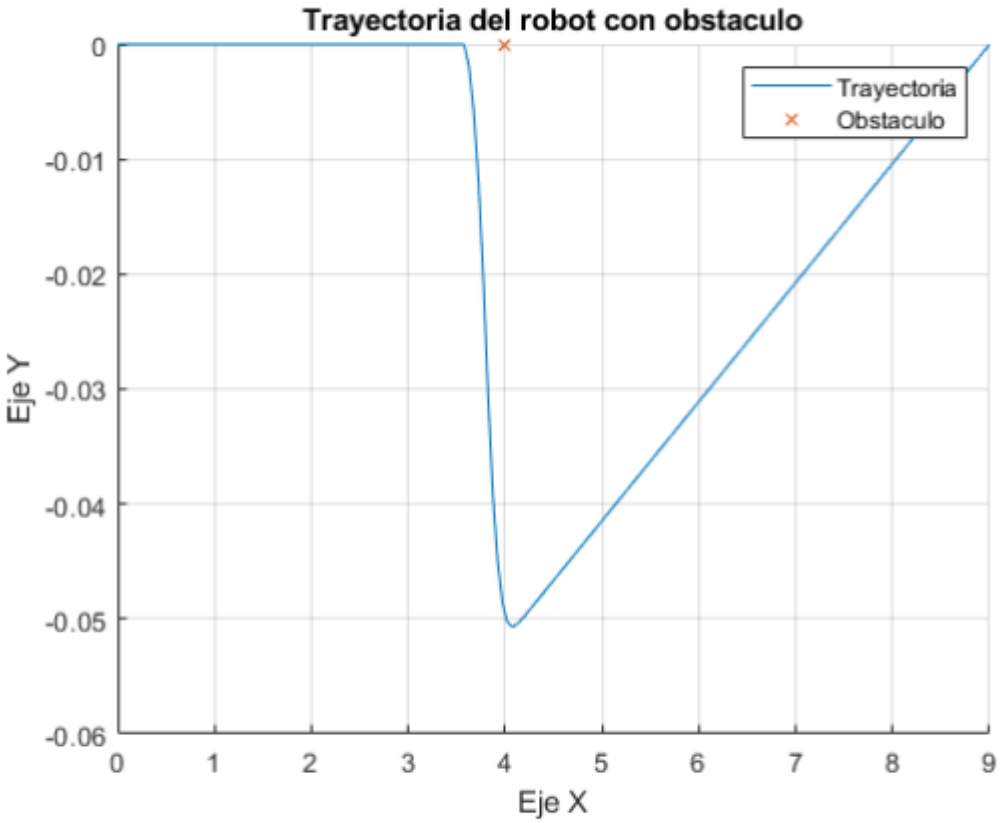


Figura 23. El robot esquiva el obstáculo.

Y, por último, una prueba de la que necesita bastante precisión para esquivarlo con las variables de refx = 5, refy = 5, obsx = 4.6 y obsy=4.6.

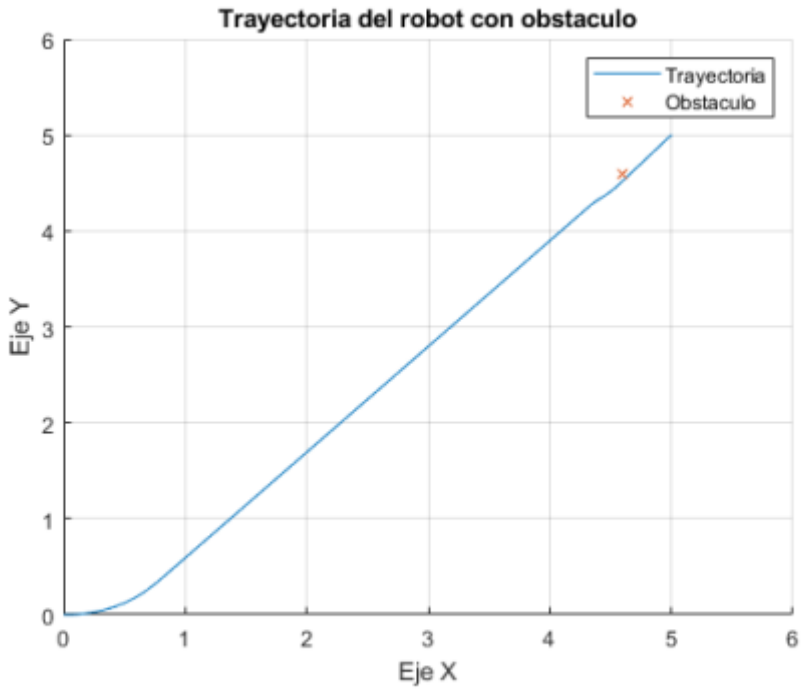


Figura 24. El robot parece que roza el obstáculo.

En principio parece que llega a rozar el obstáculo, pero si se hace zoom a la gráfica se puede observar que lo esquiva con éxito.

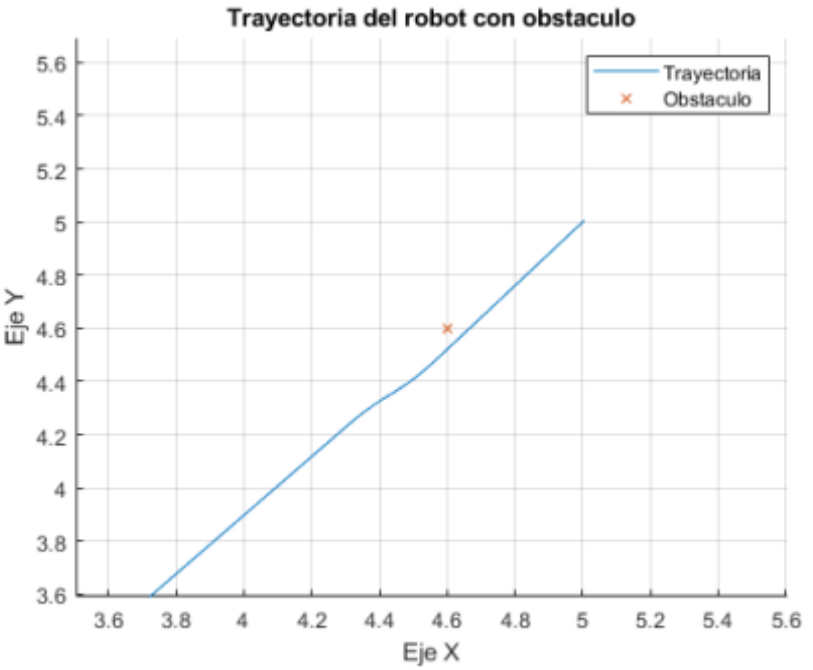


Figura 25. Con zoom se ve que el robot esquiva con éxito el obstáculo.

.