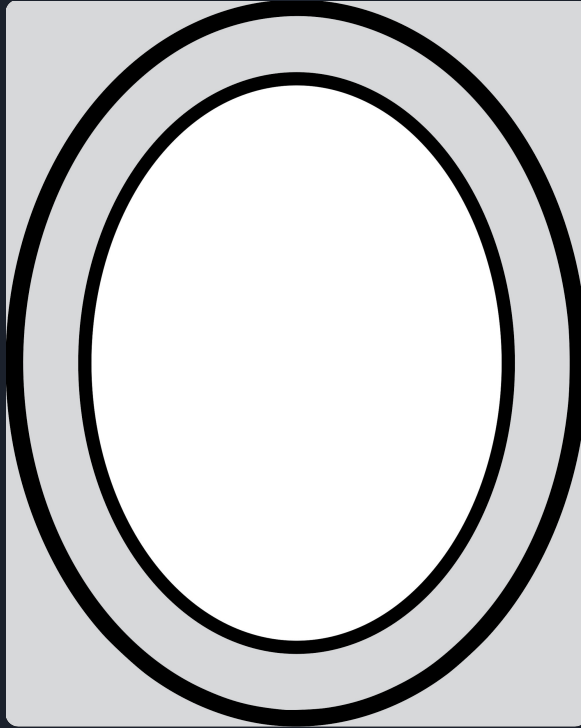


Práctica Final. Diseño de Controladores Borrosos y Neuroborrosos para un Robot Móvil

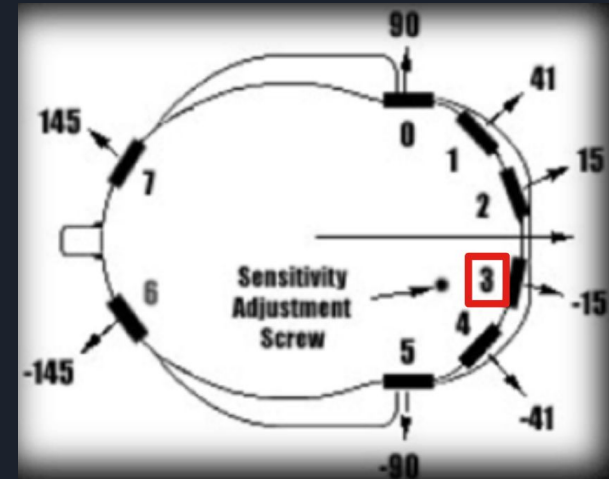
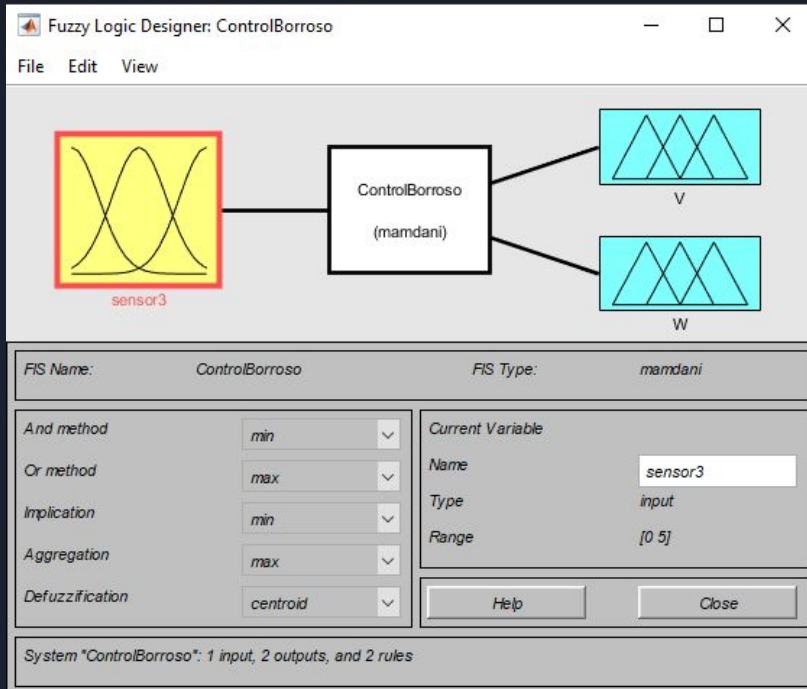
Ana Cortés Cercadillo
Carlos Javier Hellín Asensio



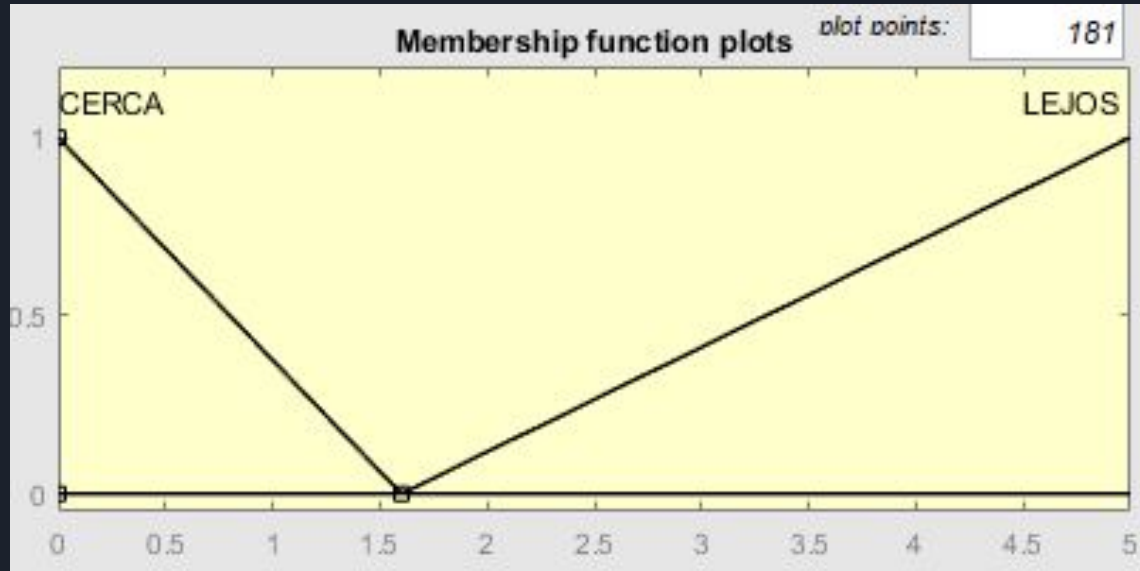
Parte 1. Control borroso. Entorno sin obstáculos.



Parte 1. Control borroso. Entorno sin obstáculos.

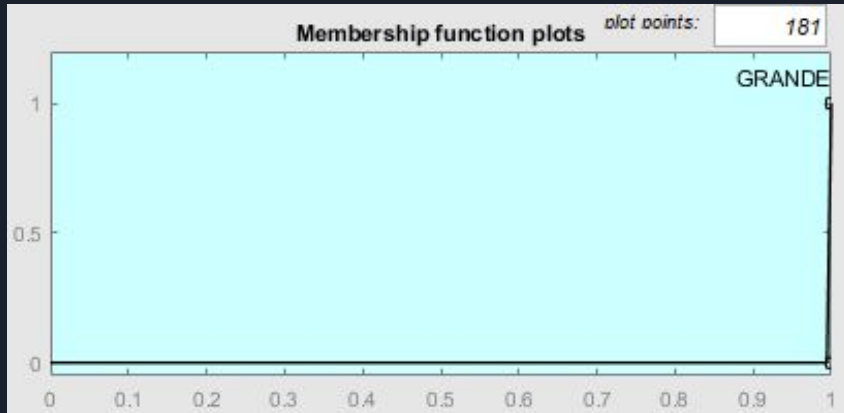


Parte 1. Control borroso. Entorno sin obstáculos.

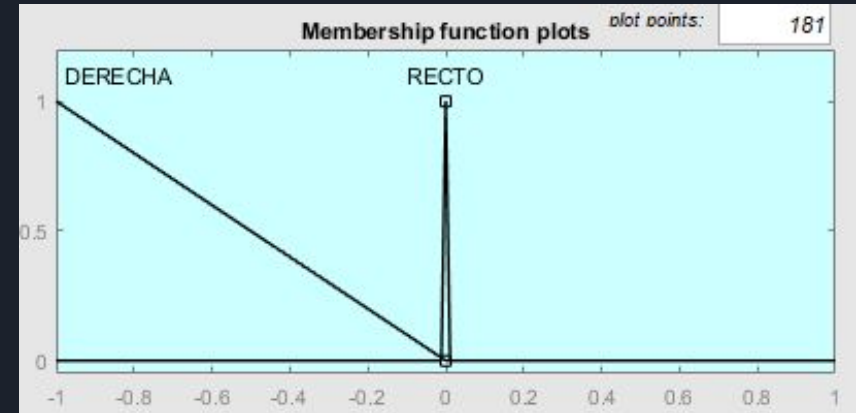


sensor3

Parte 1. Control borroso. Entorno sin obstáculos.



V



W

Parte 1. Control borroso. Entorno sin obstáculos.

Rule Editor: ControlBorrosoSO

File Edit View Options

1. If (sensor3 is LEJOS) then (V is GRANDE)(W is DERECHA) (1)
2. If (sensor3 is CERCA) then (V is GRANDE)(W is RECTO) (1)

If sensor3 is

CERCA
LEJOS
none

☐ not

Then V is and W is

GRANDE
none

RECTO
DERECHA
none

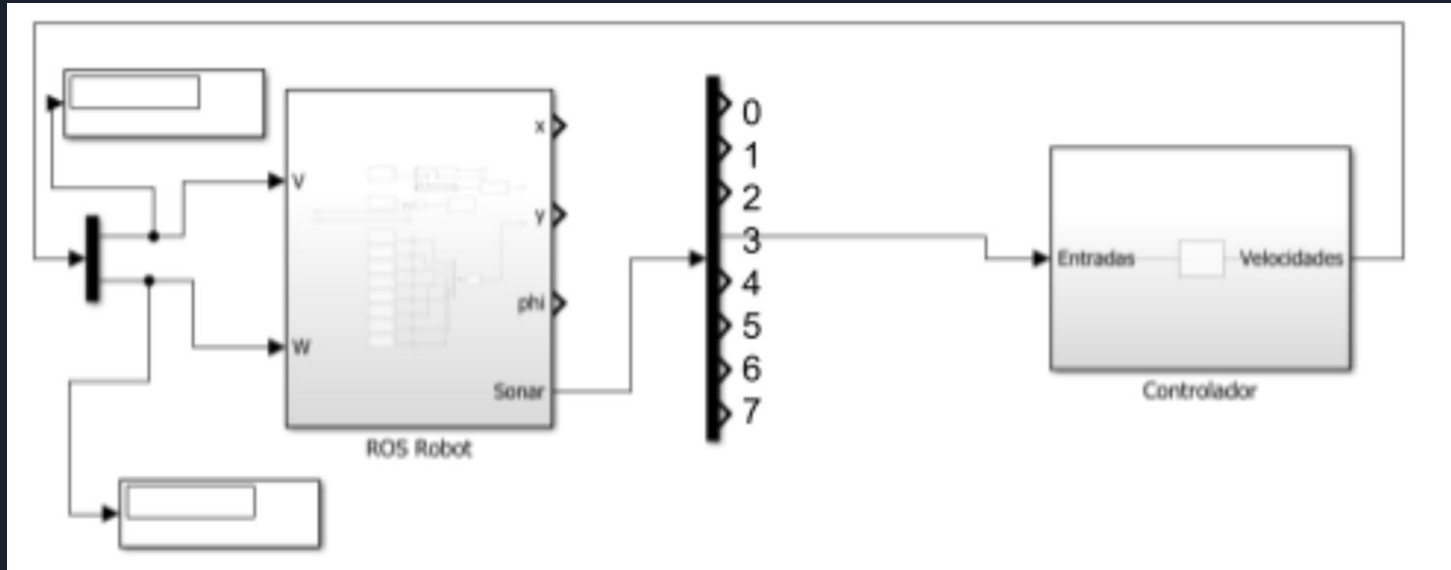
☐ not ☐ not

Connection: ☐ or ☒ and Weight: 1

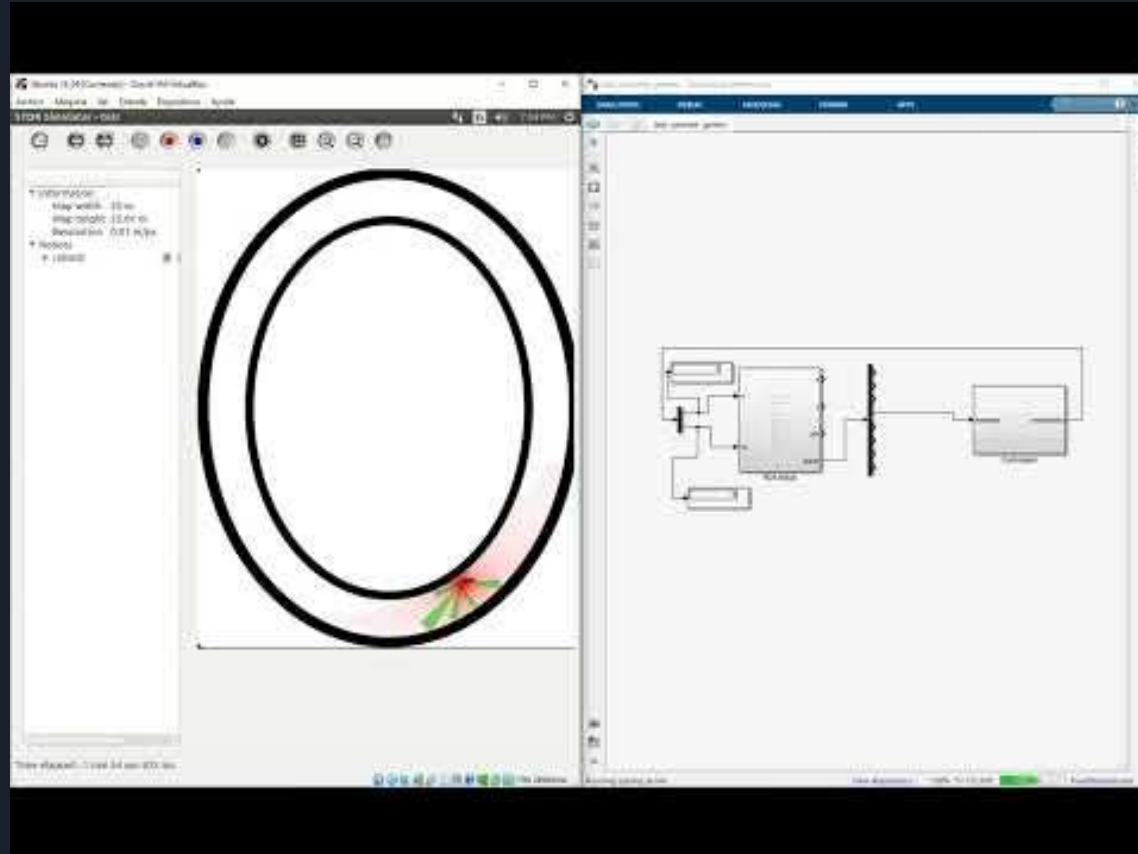
Delete rule Add rule Change rule << >>

FIS Name: ControlBorrosoSO Help Close

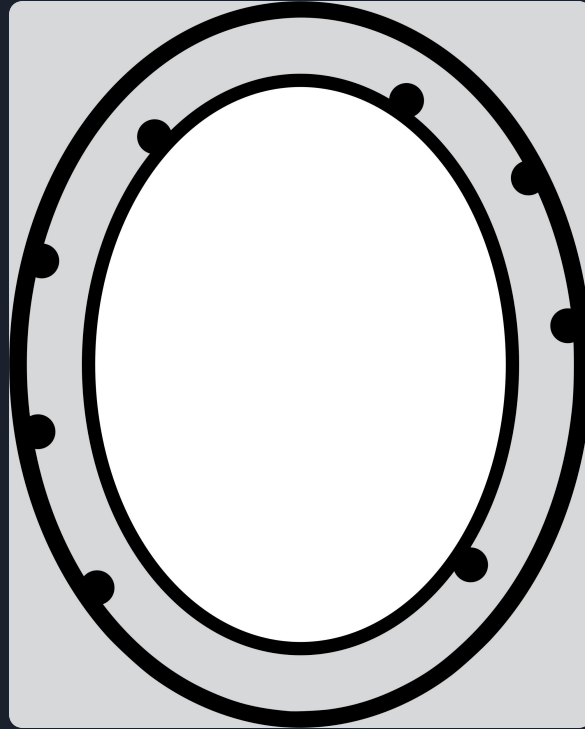
Parte 1. Control borroso. Entorno sin obstáculos.



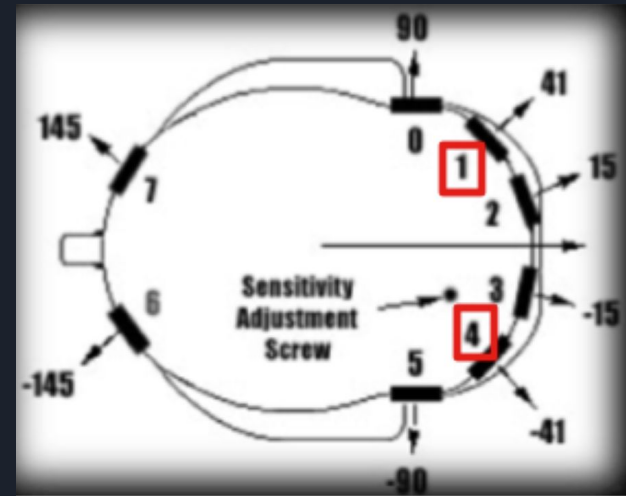
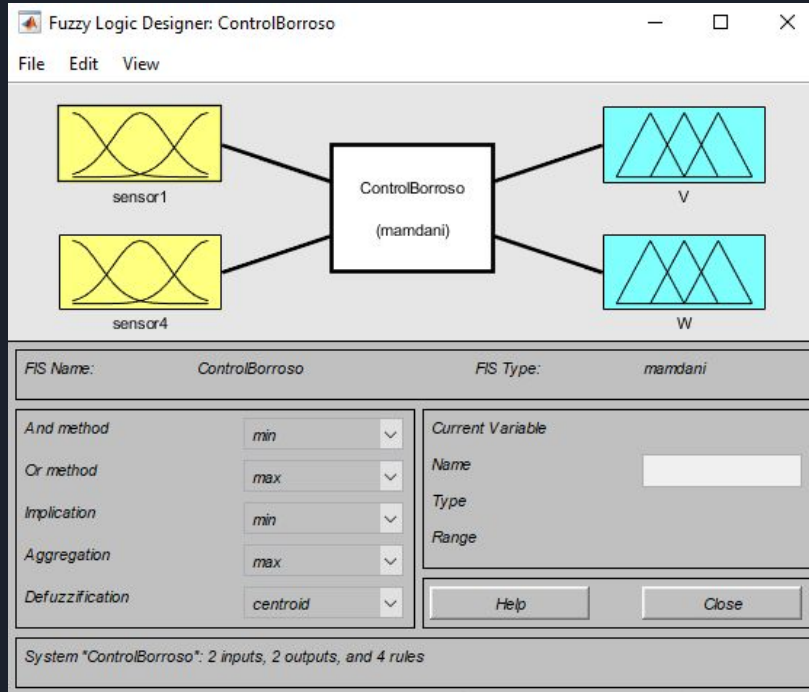
Parte 1. Control borroso. Entorno sin obstáculos.



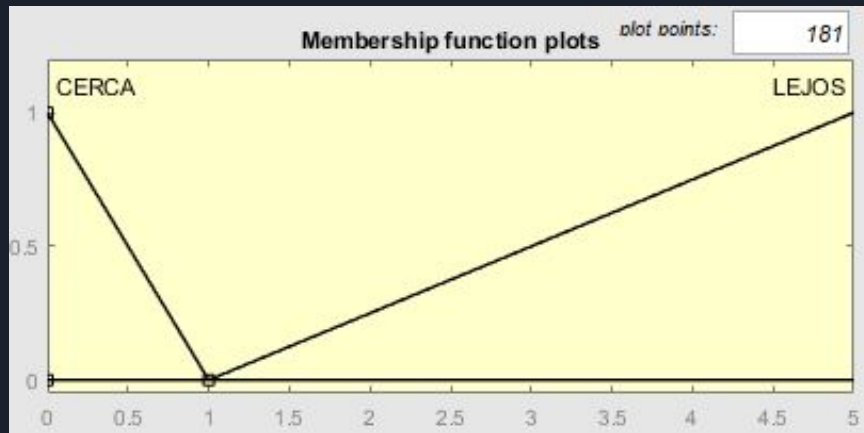
Parte 1. Control borroso. Entorno con obstáculos.



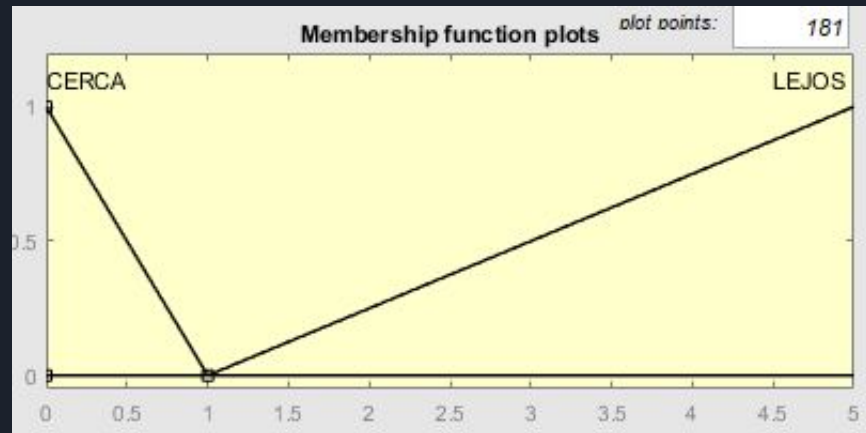
Parte 1. Control borroso. Entorno con obstáculos.



Parte 1. Control borroso. Entorno con obstáculos.

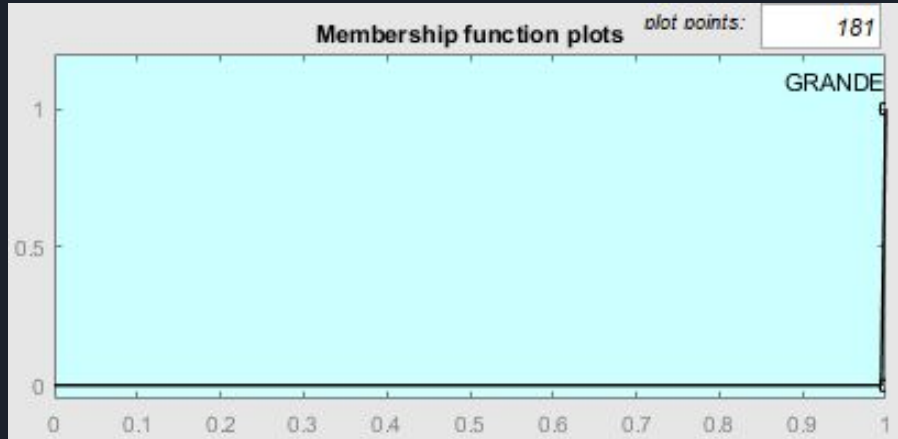


sensor1

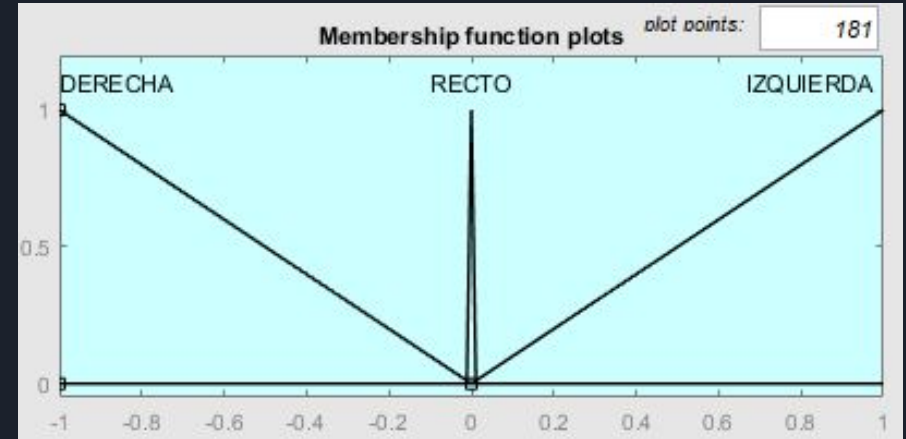


sensor4

Parte 1. Control borroso. Entorno con obstáculos.



V



W

Parte 1. Control borroso. Entorno con obstáculos.

Rule Editor: ControlBorroso

File Edit View Options

1. If (sensor1 is CERCA) and (sensor4 is LEJOS) then (V is GRANDE)(W is DERECHA) (1)
2. If (sensor1 is LEJOS) and (sensor4 is CERCA) then (V is GRANDE)(W is IZQUIERDA) (1)
3. If (sensor1 is LEJOS) and (sensor4 is LEJOS) then (V is GRANDE)(W is RECTO) (1)
4. If (sensor1 is CERCA) and (sensor4 is CERCA) then (V is GRANDE)(W is RECTO) (1)

If sensor1 is CERCA LEJOS none

and sensor4 is CERCA LEJOS none

Then V is GRANDE none

and W is DERECHA RECTO IZQUIERDA none

☐ not ☐ not ☐ not ☐ not

Connection ☐ or ☒ and

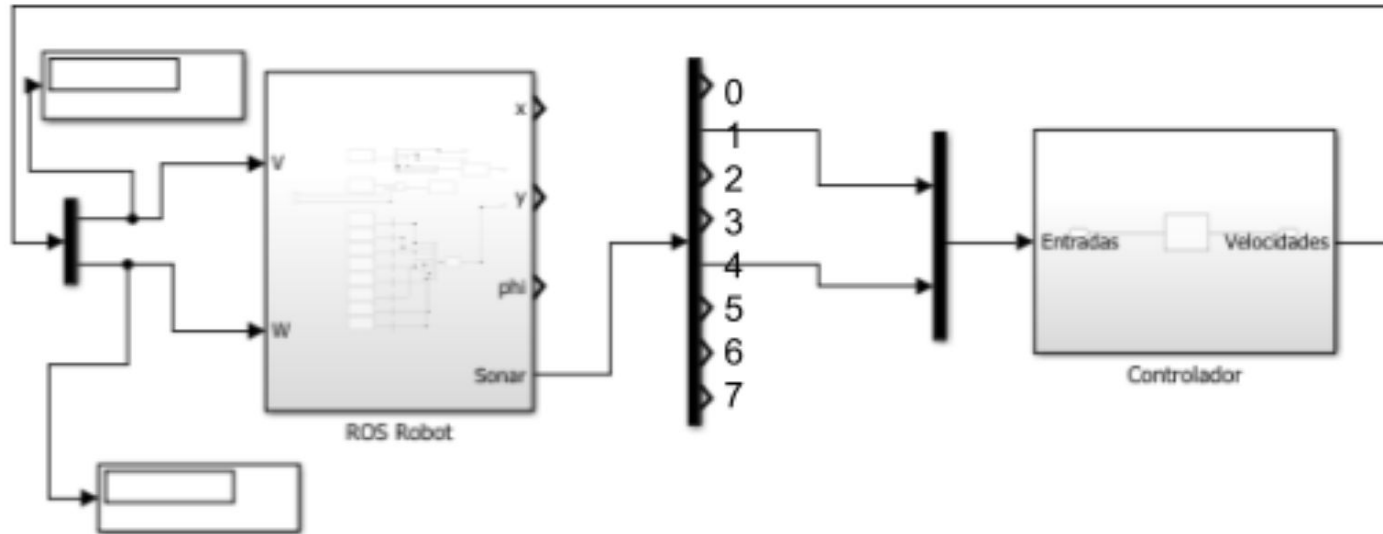
Weight: 1

Delete rule Add rule Change rule << >>

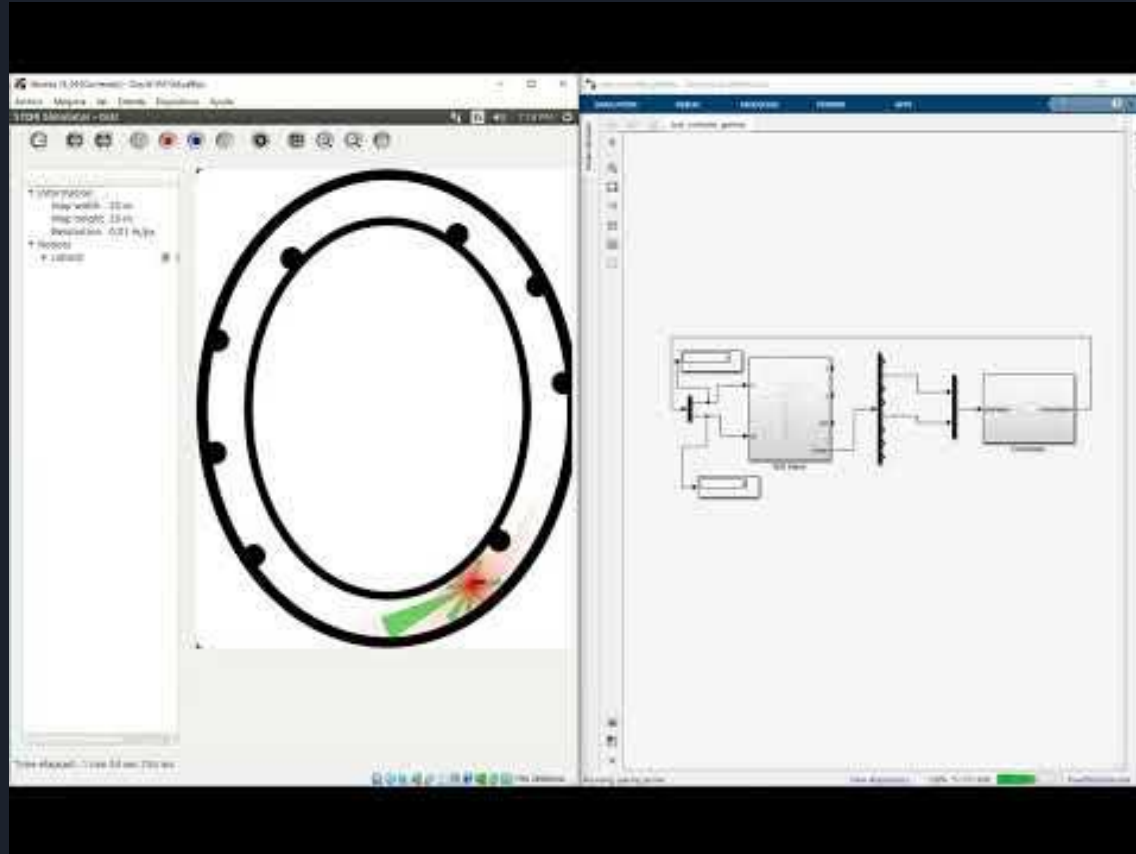
FIS Name: ControlBorroso

Help Close

Parte 1. Control borroso. Entorno con obstáculos.



Parte 1. Control borroso. Entorno con obstáculos.

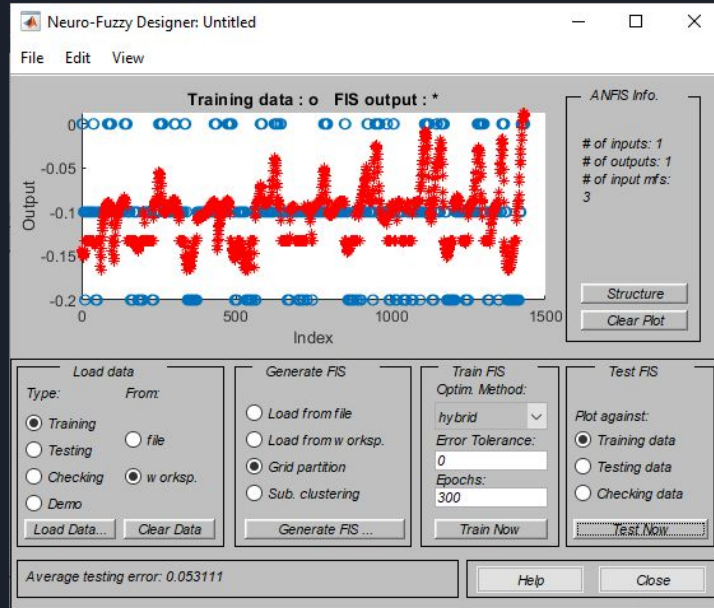


Parte 2. Controlador Neuroborroso. Entorno sin obstáculos.

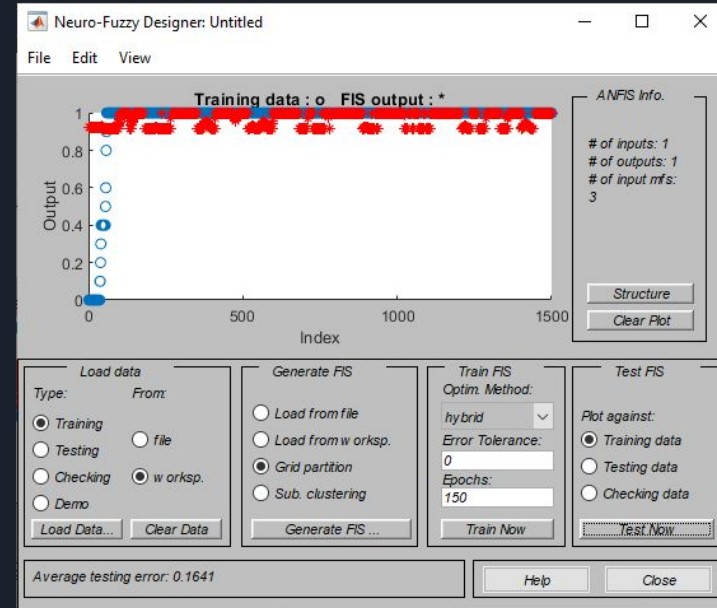
```
Train_angular.m  x  +  
1 - train_angular = training(:,[4,12])  
2 - indices = round(linspace(1,size(training,1),1500))  
3 - train_angular = train_angular(indices,:)  
4 - train_angular(isinf(train_angular)) = 5.0  
5 - train_angular = double(train_angular)
```

```
Train_lineal.m  x  +  
1 - train_lineal = training(:,[4,13])  
2 - indices = round(linspace(1,size(training,1),1500))  
3 - train_lineal = train_lineal(indices,:)  
4 - train_lineal(isinf(train_lineal)) = 5.0  
5 - train_lineal = double(train_lineal)
```


Parte 2. Controlador Neuroborroso. Entorno sin obstáculos.

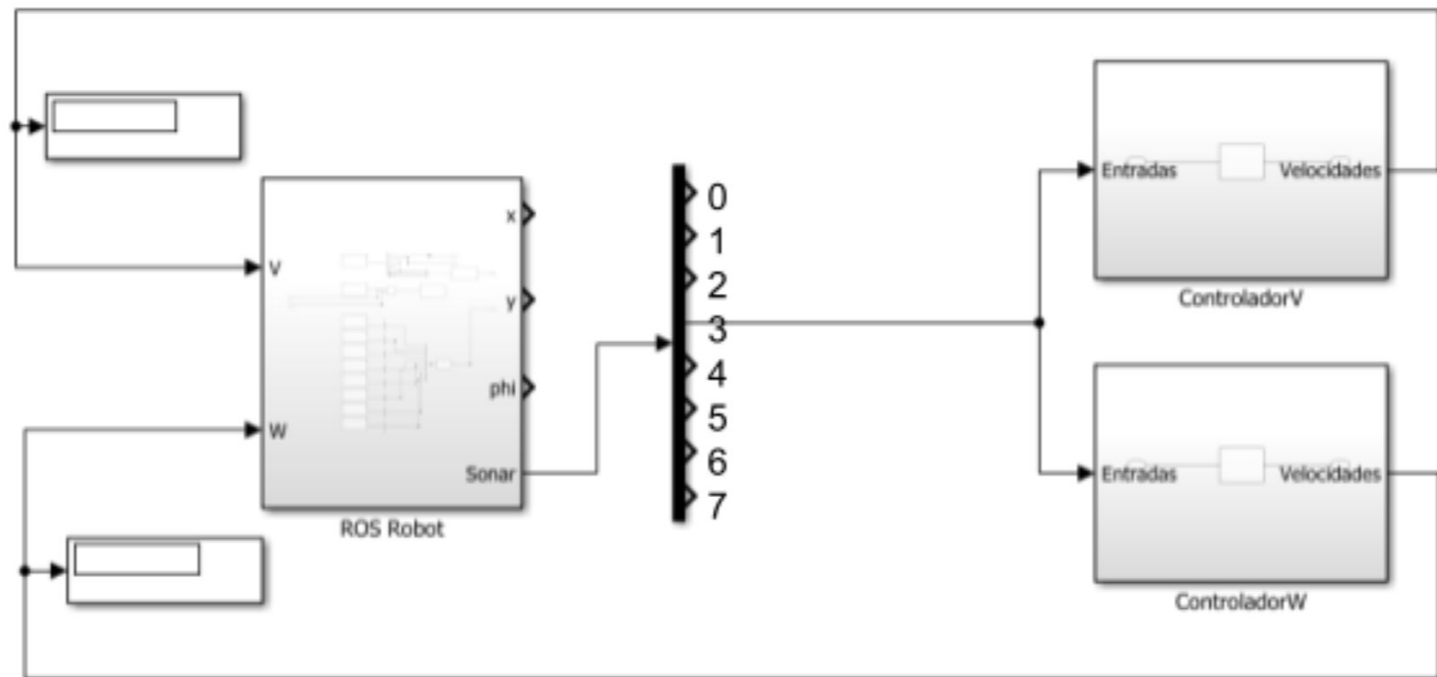


W

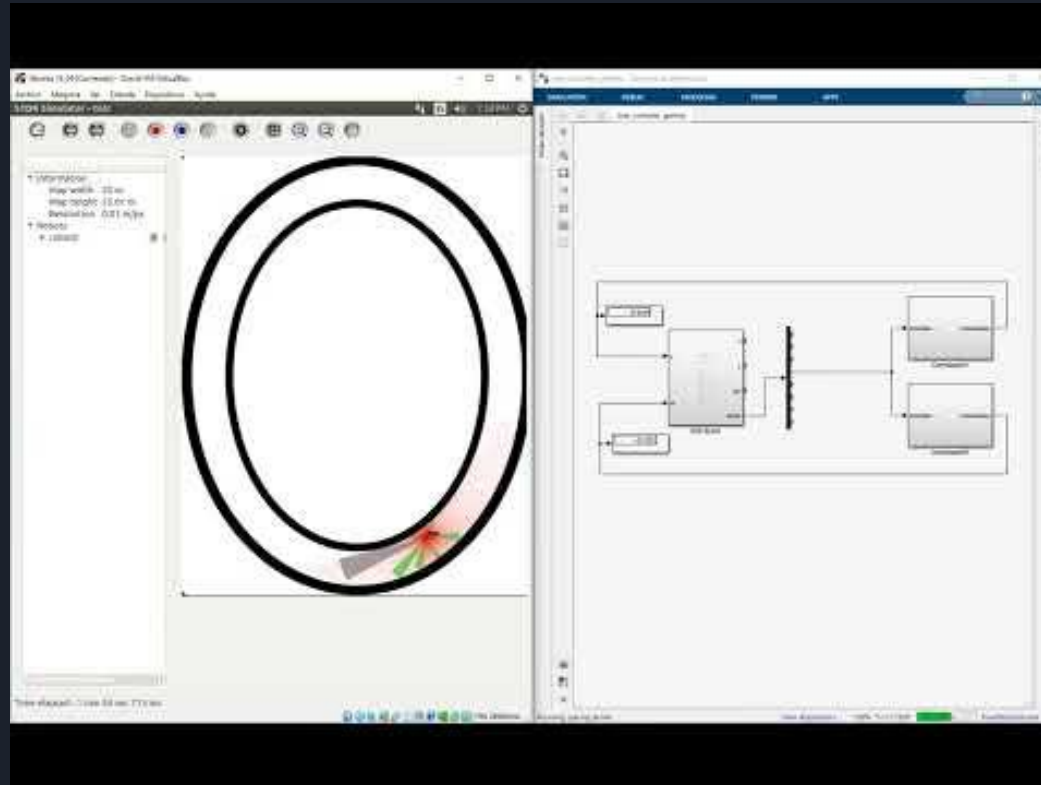


V

Parte 2. Controlador Neuroborroso. Entorno sin obstáculos.



Parte 2. Controlador Neuroborroso. Entorno sin obstáculos.

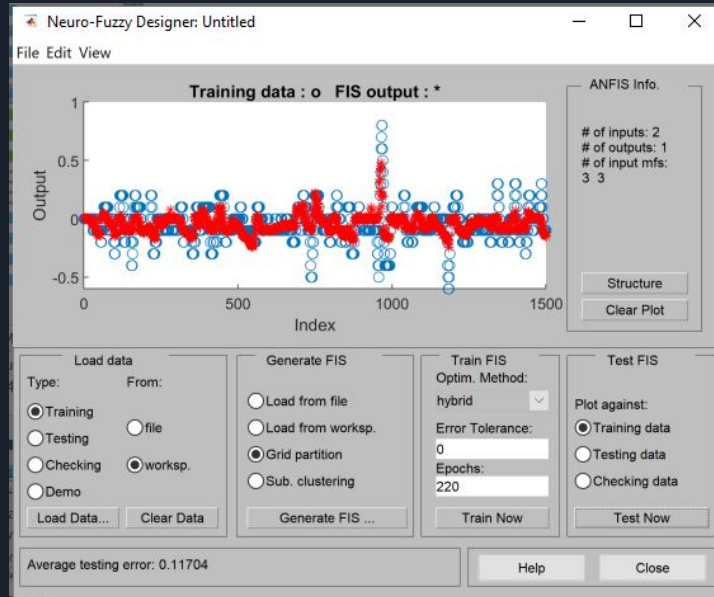


Parte 2. Controlador Neuroborroso. Entorno con obstáculos.

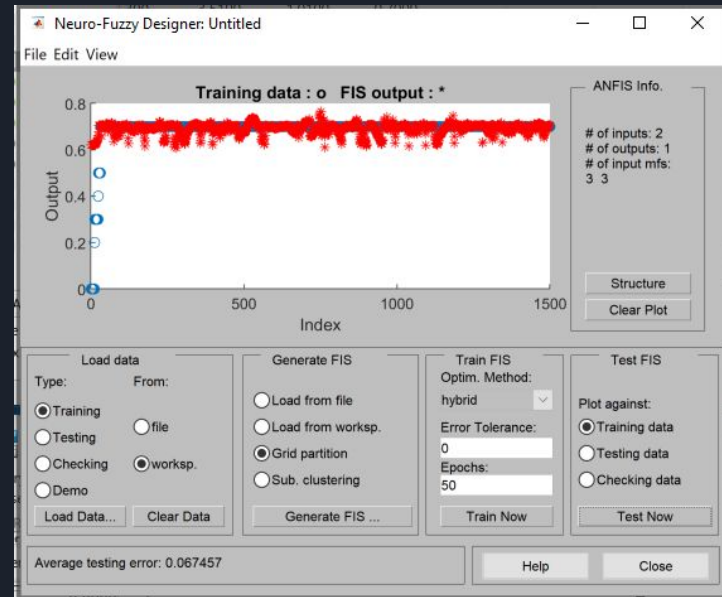
```
Train_angular.m  x  +
1 -   train_angular = training(:, [3,4,12])
2 -   indices = round(linspace(1,size(training,1),1500))
3 -   train_angular = train_angular(indices,:)
4 -   train_angular(isinf(train_angular)) = 5.0
5 -   train_angular = double(train_angular)
```

```
Train_lineal.m  x  +
1 -   train_lineal = training(:, [3,4,13])
2 -   indices = round(linspace(1,size(training,1),1500))
3 -   train_lineal = train_lineal(indices,:)
4 -   train_lineal(isinf(train_lineal)) = 5.0
5 -   train_lineal = double(train_lineal)
```

Parte 2. Controlador Neuroborroso. Entorno con obstáculos.

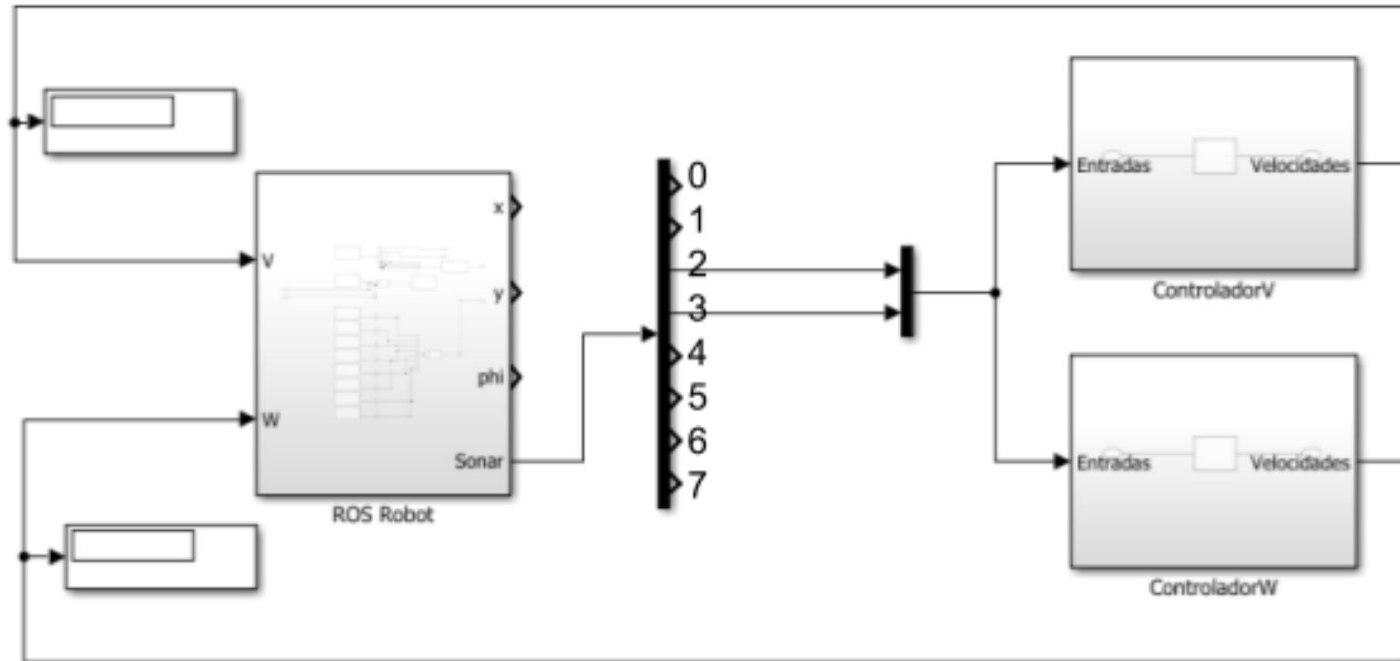


W

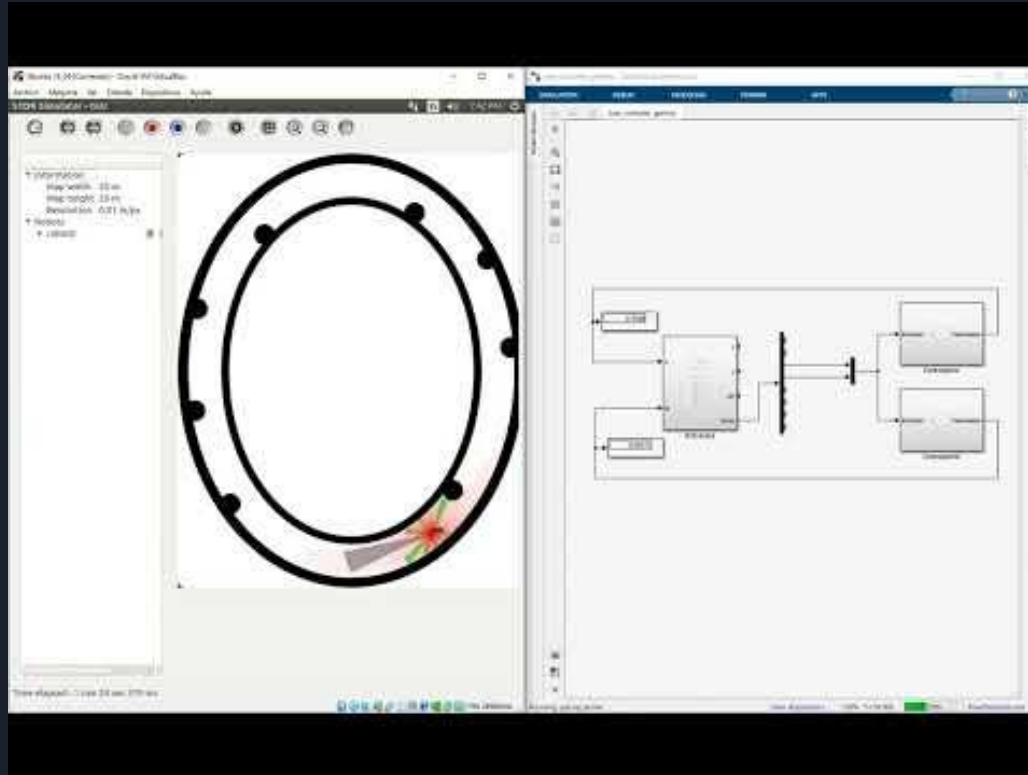


V

Parte 2. Controlador Neuroborroso. Entorno con obstáculos.



Parte 2. Controlador Neuroborroso. Entorno con obstáculos.



¡¡Gracias por todo!!

