



PRÁCTICA 2. CONTROL BORROSO

SISTEMAS DE CONTROL INTELIGENTE

Álvaro Zamorano Ortega, Gabriel López Cuenca

Contenido

PARTE 1 2

 Controlador de posición a un punto de referencia 2

 Seguimiento de trayectoria..... 7

PARTE 2 10

 Controlador de posición a un punto de referencia 10

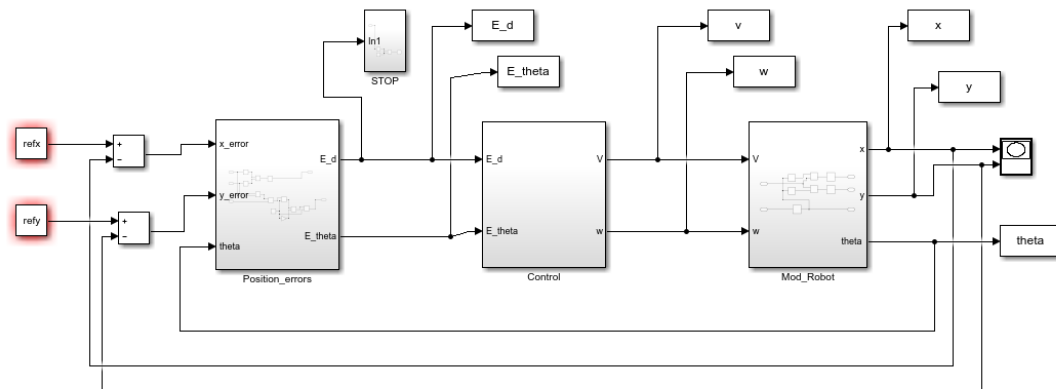
 Seguimiento de trayectoria..... 12

PARTE 1

Controlador de posición a un punto de referencia

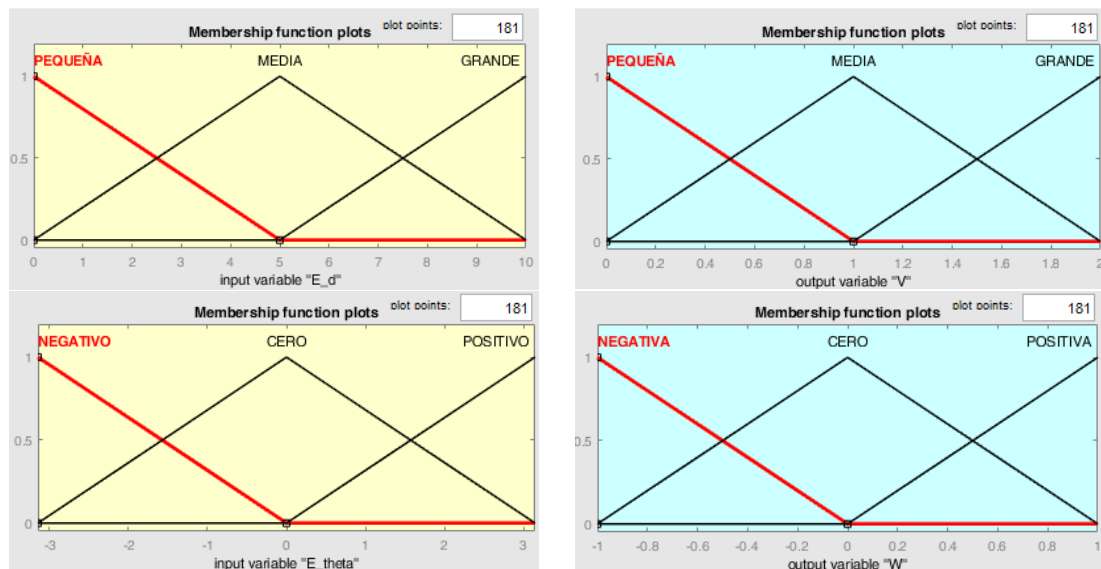
Se han creado diferentes tipos de controladores borrosos mediante la herramienta fuzzy hasta encontrar uno capaz de llegar a la máxima cantidad de puntos indicados con un error de **0.01 metros**.

Para comenzar, el esquema creado en **Simulink** es el siguiente:



En él destaca el bloque **“Control”** ya que es donde se introducirán los diferentes controladores creados.

En primer lugar, se usaron unas **funciones de pertenencia** similares a las vistas en clase de teoría:

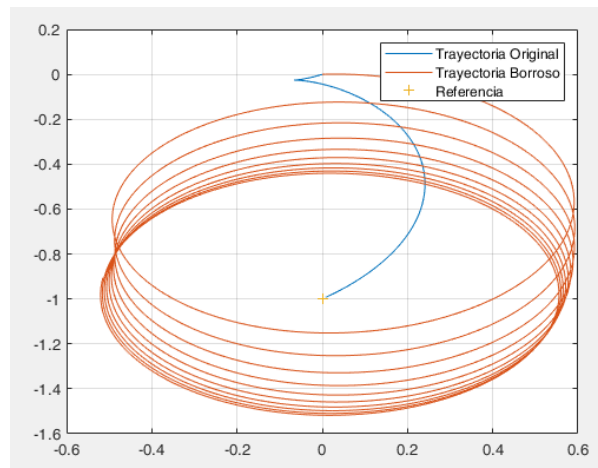


Y una tabla de reglas:

1. If (E_d is GRANDE) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
2. If (E_d is GRANDE) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)
3. If (E_d is PEQUEÑA) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
4. If (E_d is PEQUEÑA) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)
5. If (E_d is MEDIA) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
6. If (E_d is MEDIA) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)

7. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
8. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
9. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
10. If (E_theta is NEGATIVO) then (W is NEGATIVA) (1)
11. If (E_theta is POSITIVO) then (W is POSITIVA) (1)

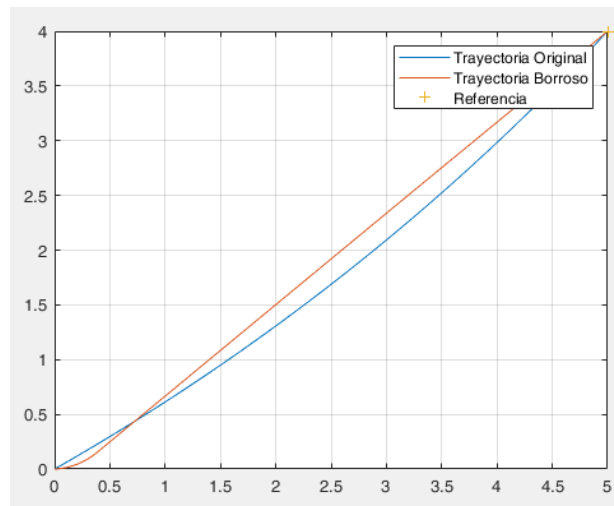
Pero observamos que el comportamiento del robot **no era correcto**, sobre todo cuando el punto de referencia se encontraba sobre el eje y, y la orientación del mismo era hacía el eje x. Cuando dicho punto era del estilo **(0,-1)** la trayectoria seguida era la siguiente:



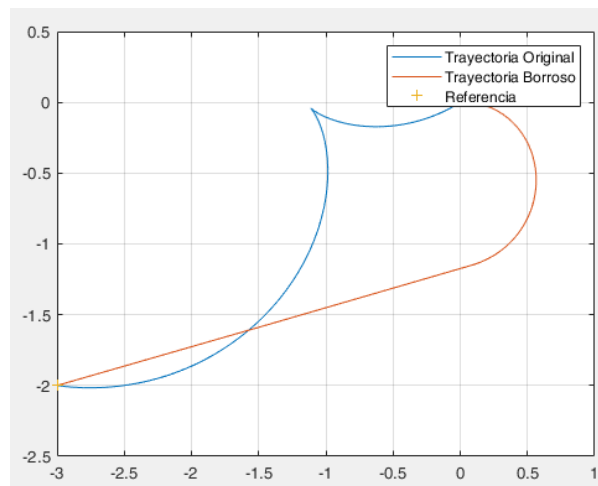
Es decir, el robot no era capaz de alcanzar el punto indicado. Esto se debía a que la velocidad lineal de salida era bastante grande y, por lo tanto, el robot no era capaz de girar lo necesario en un rango de distancia tan corto.

Cabe destacar que en este caso si se conseguía llegar a puntos más alejados, como, por ejemplo:

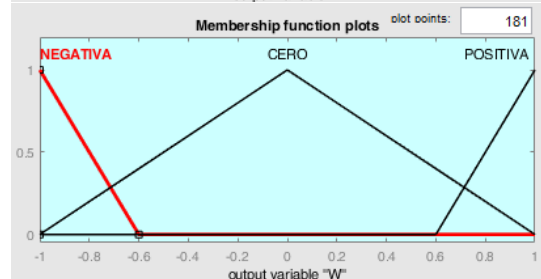
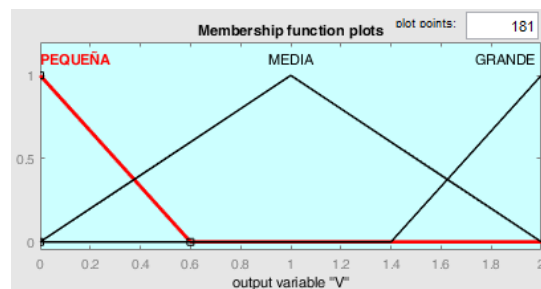
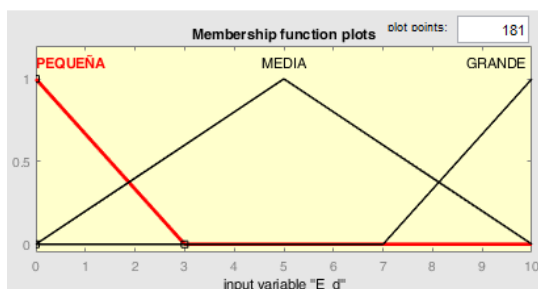
Referencia = **(5,4)**



Referencia = **(-3,-2)**

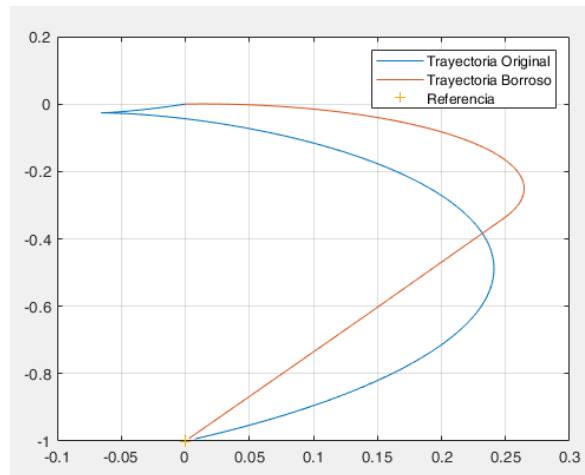


Posteriormente modificamos las **funciones de pertenencia** de cada conjunto, así como la tabla de reglas. En esta ocasión tanto la variable de entrada E_d como las de la salida V y W redujeron sus intervalos de pertenencia en sus conjuntos extremos, es decir, en pequeña, grande, negativa y positiva respectivamente.

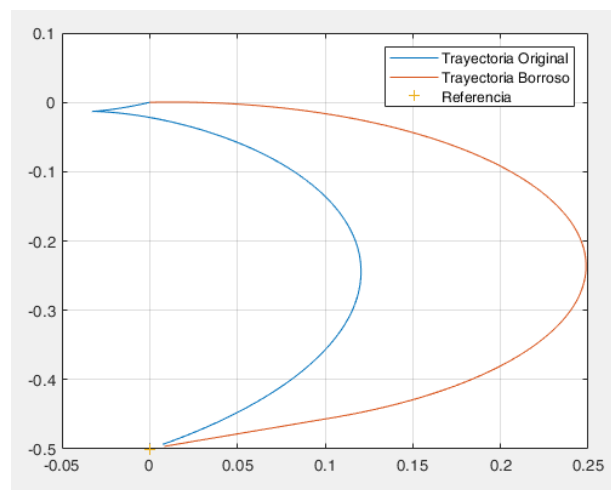


Y el cambio realizado en la **tabla de reglas** fue indicar que la **velocidad lineal** en todo momento fuese pequeña, independientemente del valor de las variables de entrada. Con esto se consiguió que el robot fuese mucho más ágil a la hora de realizar giros en un pequeño espacio. En el resto de las ocasiones el comportamiento era similar.

En este caso se puede apreciar cómo es capaz de **alcanzar** el punto al que anteriormente no conseguía llegar (0,-1), ya que la velocidad lineal de salida es menor y por lo tanto el rango de giro aumenta.



Incluso es capaz de llegar a puntos tan cercanos en y de sí mismo como puede ser el **(0,-0.5)**.

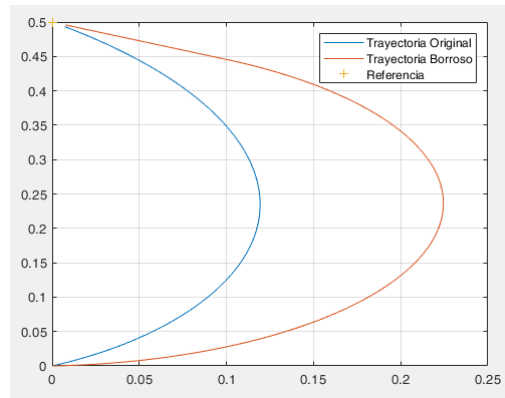


Por último, observamos que un gran número de **reglas** eran insignificantes por lo que la tabla de reglas se redujo y ha quedado de la siguiente manera:

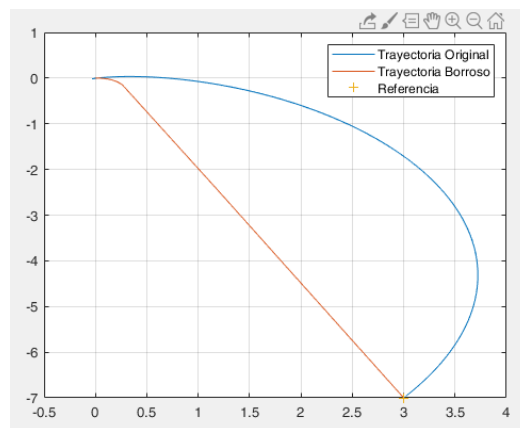
1. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
2. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
3. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
4. If (E_theta is NEGATIVO) then (W is NEGATIVA) (1)
5. If (E_theta is POSITIVO) then (W is POSITIVA) (1)

En este caso el funcionamiento del controlador no se ve afectado. Ejemplos de ello son:

Referencia = **(0,0.5)**



Referencia = **(3,-7)**

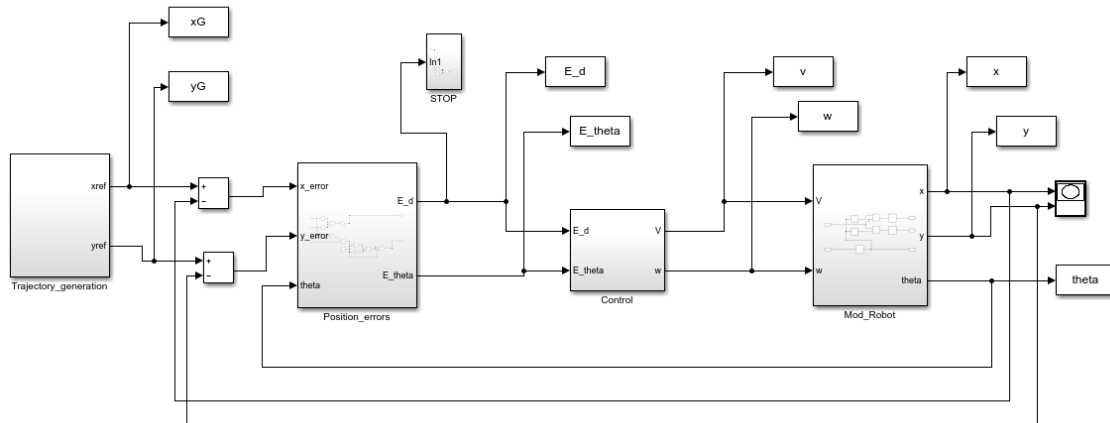


En todos los ejemplos se ha incluido la trayectoria que realiza el robot con el **controlador original** y se puede apreciar que la conseguida con el controlador borroso, en ocasiones, es más directa.

Es importante que la **distancia al punto referencia** sea menor de 10, es decir, menor que el rango del error de distancia (variable de entrada) ya que sino se encontrará fuera de este rango y el controlador no funcionará de forma correcta.

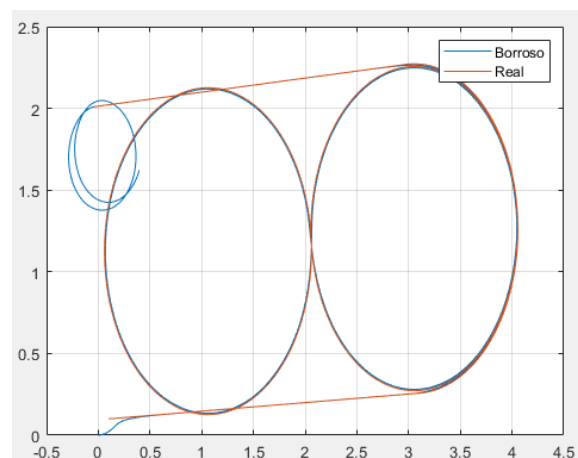
Seguimiento de trayectoria

El **diagrama** de control para seguimiento de trayectorias que hemos creado es el siguiente:



Los principales cambios que hemos realizado con respecto al diagrama de la práctica anterior son:

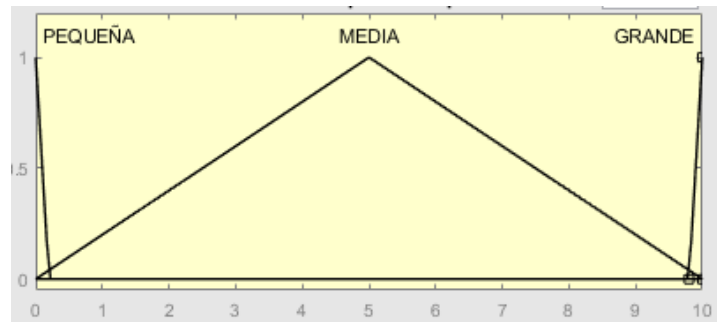
- Hemos añadido una **condición de parada** cuando el error de distancia E_d es inferior a 0.01 metros. Sin este bloque, al final de la trayectoria seguía un pequeño círculo hasta agotar el tiempo de simulación, como muestra la siguiente imagen:



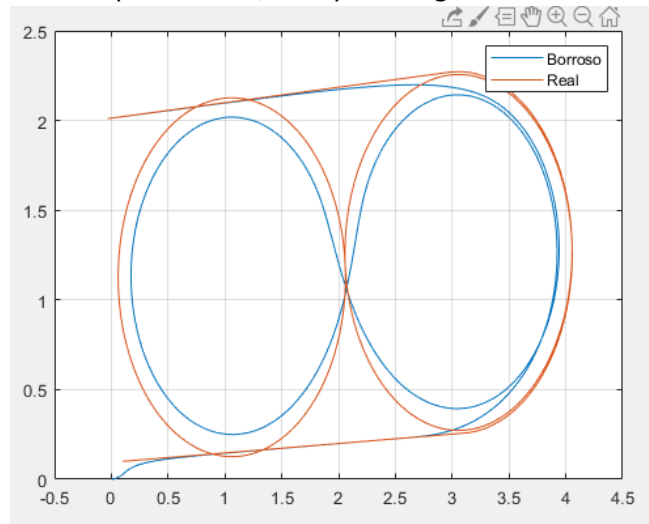
- La **posición inicial** del robot es distinta a la posición inicial de la trayectoria. Esto es debido a la condición de parada creada.

Respecto a los cambios en el diseño del controlador borroso han sido los siguientes:

- Cambio de la **función de pertenencia del error de distancia** PEQUEÑO y GRANDE (más exageradas). El cambio se debe a que ahora los puntos generados que tiene que llegar el robot se encuentran a una distancia muy pequeña. Las funciones de pertenencias son las siguientes:



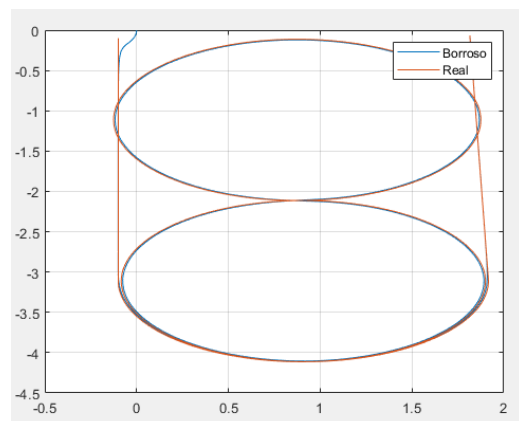
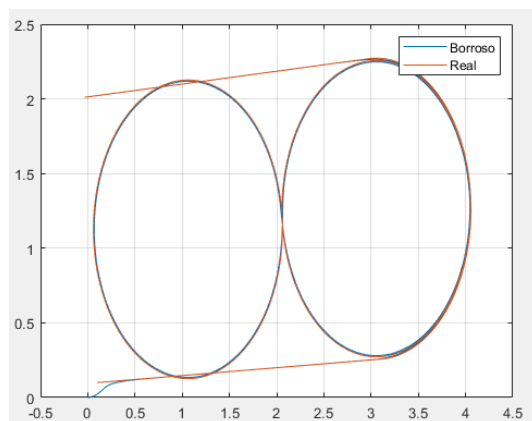
- Sin el cambio de las funciones de pertenencia, la trayectoria generada sería la siguiente:

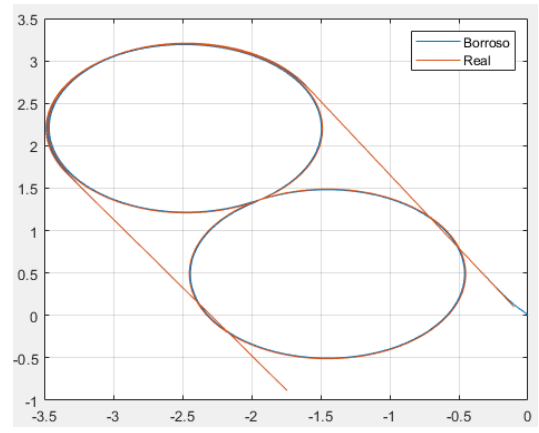
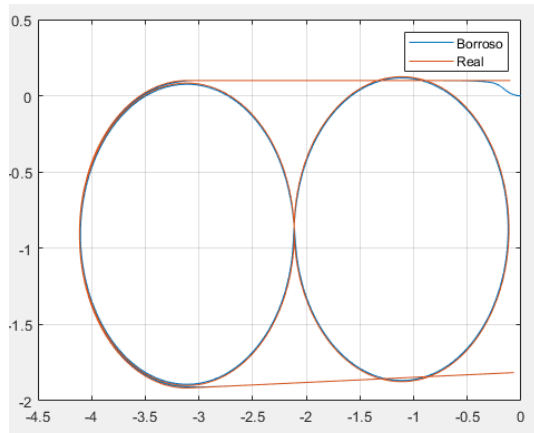


Las **reglas** del controlador borroso son iguales que anteriormente:

1. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
2. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
3. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
4. If (E_theta is NEGATIVO) then (W is NEGATIVA) (1)
5. If (E_theta is POSITIVO) then (W is POSITIVA) (1)

Con todo ello, las siguientes imágenes muestran **diferentes trayectorias** en donde se puede observar una correcta simulación:

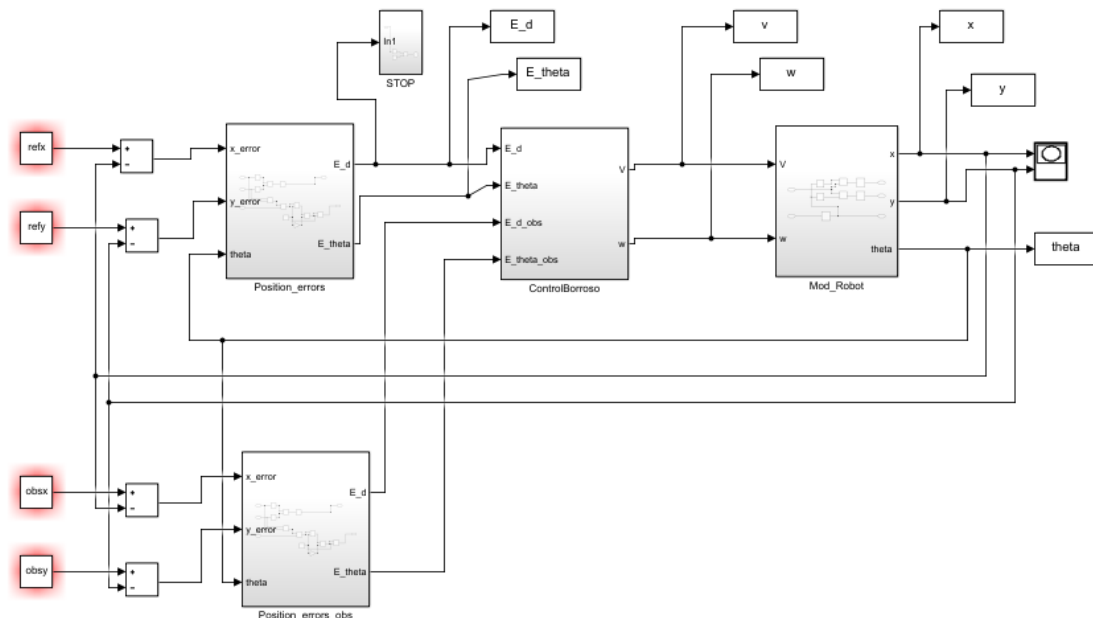




PARTE 2

Controlador de posición a un punto de referencia

A partir del diagrama anterior se han introducido bloques para controlar la distancia y el ángulo al que nos encontramos del **objeto**:

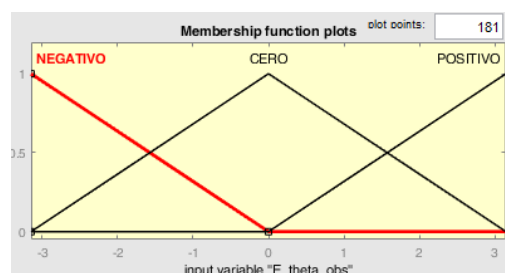
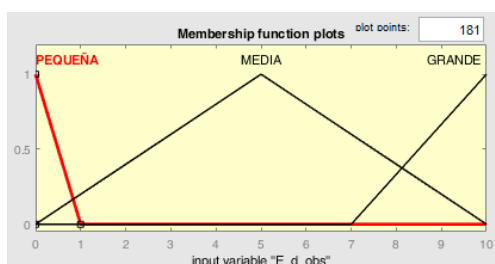


Respecto al controlado borroso, este ahora contiene dos nuevas entradas que son **E_d_obs** y **E_theta_obs**. Al igual que anteriormente en primer lugar se probaron diferentes controladores. El primero de ellos tenía las siguientes características:

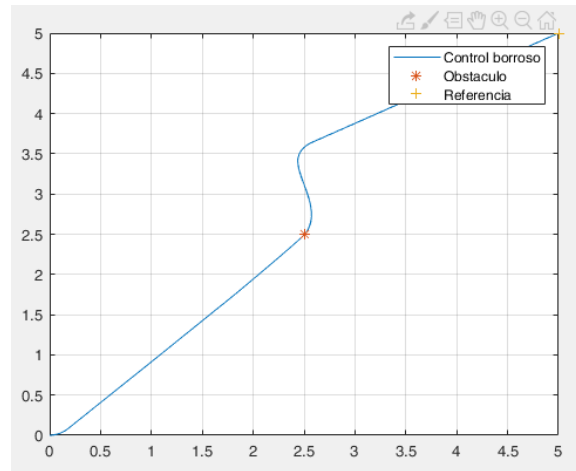
Tabla de reglas:

1. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
2. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
3. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
4. If (E_theta is NEGATIVO) then (W is NEGATIVA) (1)
5. If (E_theta is POSITIVO) then (W is POSITIVA) (1)
6. If (E_d_obs is PEQUEÑA) and (E_theta_obs is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)
7. If (E_d_obs is PEQUEÑA) and (E_theta_obs is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
8. If (E_d is GRANDE) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
9. If (E_d is GRANDE) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)
10. If (E_d is PEQUEÑA) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
11. If (E_d is PEQUEÑA) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)
12. If (E_d is MEDIA) and (E_theta is NEGATIVO) then (V is PEQUEÑA)(W is NEGATIVA) (1)
13. If (E_d is MEDIA) and (E_theta is POSITIVO) then (V is PEQUEÑA)(W is POSITIVA) (1)

Las **funciones de pertenencia** son igual que anteriormente, pero introduciendo las de las dos nuevas variables de entrada:

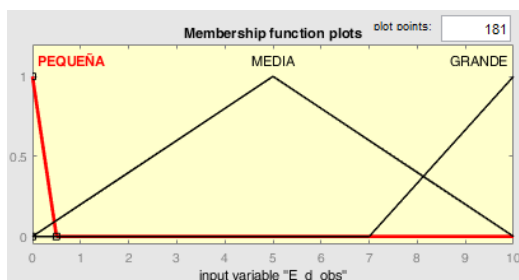


De esta forma **no** se conseguía esquivar el obstáculo a tiempo ya que cuando se detectaba el objeto cercano, el robot no era capaz de girar a tiempo, ejemplo de ello:



Posteriormente se modificaron la **función de pertenencia de E_{d_obs}** llevando más el extremo el conjunto "PEQUEÑA". Además, como pasaba en el caso anterior, observamos que gran cantidad de las **reglas** empleadas no eran necesarias para conseguir nuestro fin.

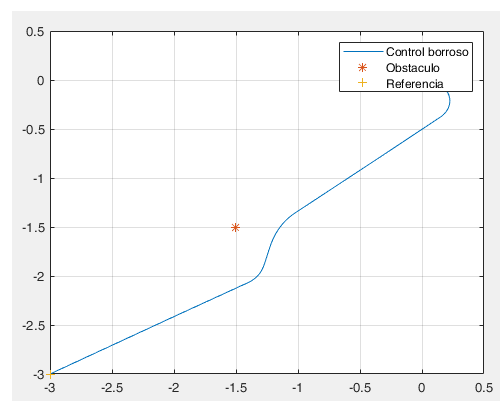
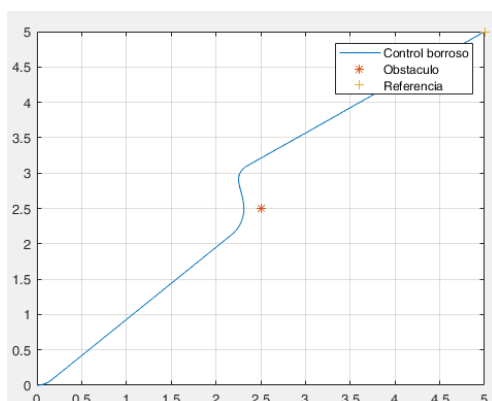
Una vez realizados los cambios, el controlador quedaba de la siguiente manera:



1. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
2. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
3. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
4. If (E_{θ} is NEGATIVO) then (W is NEGATIVA) (1)
5. If (E_{θ} is POSITIVO) then (W is POSITIVA) (1)
6. If (E_{d_obs} is PEQUEÑA) then (W is POSITIVA) (1)

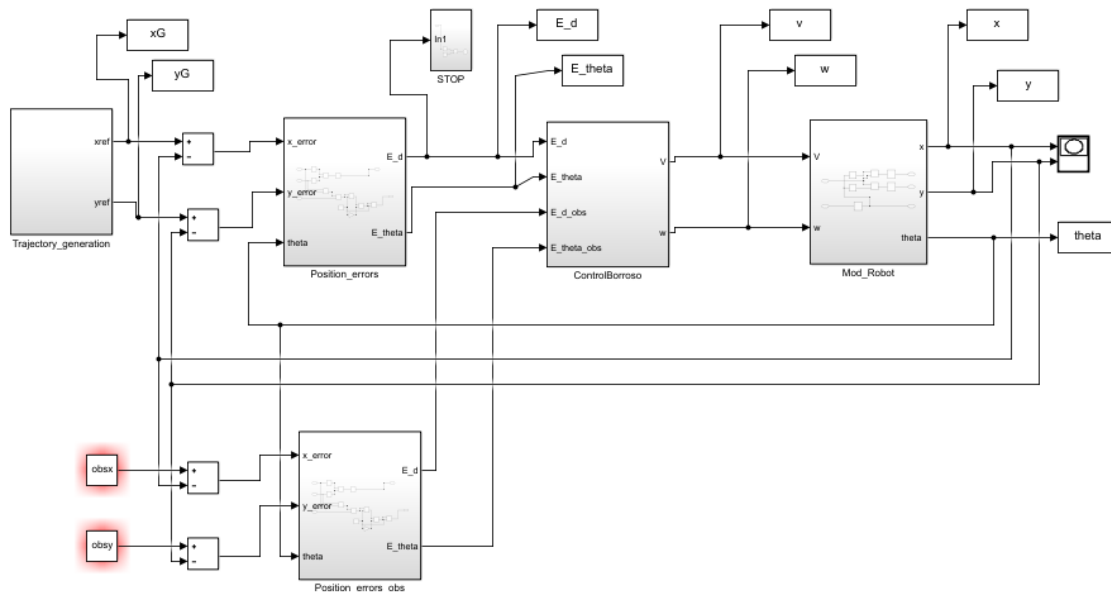
Como se puede observar, respecto a la parte 1, únicamente se ha introducido una regla haciendo referencia al obstáculo. En caso de que la distancia a este sea pequeña, el robot se girará a la **izquierda**, independientemente de hacia donde esté el objeto o hacia donde queramos llegar. Se ha realizado de esta forma ya que únicamente se indica que se debe evitar el obstáculo, sin necesidad de realizar, por ejemplo, el camino más corto.

Ejemplos de que su **funcionamiento** es correcto:



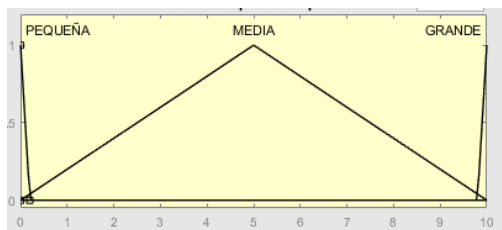
Seguimiento de trayectoria

El **diagrama** de control para seguimiento de trayectorias creado es el siguiente:



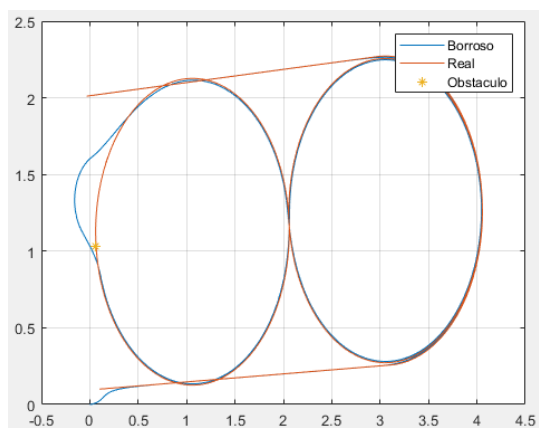
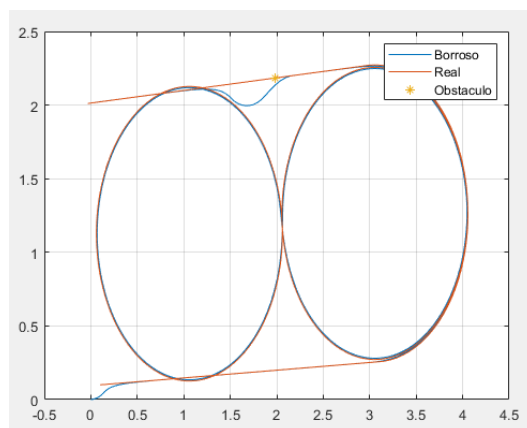
Al igual que en la parte 1, hemos añadido una **condición de parada** y que los puntos iniciales del robot y de la trayectoria sean distintos.

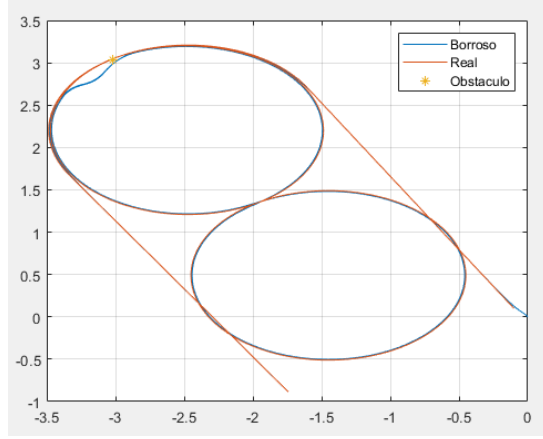
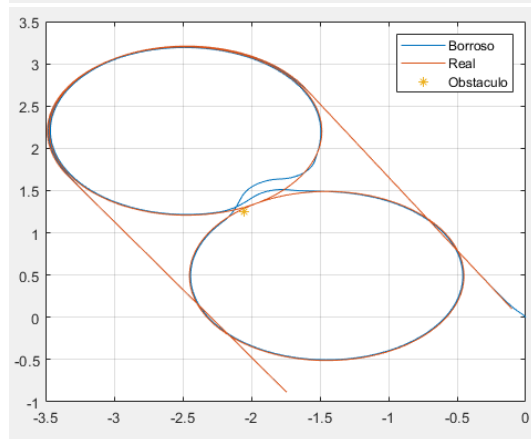
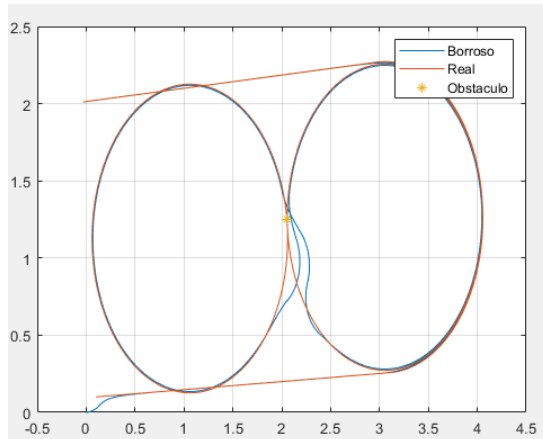
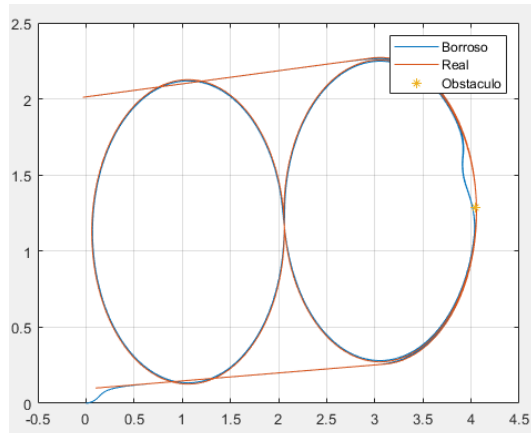
En cuanto al controlador borroso, las **reglas** son las mismas que las anteriores. Sin embargo, hemos modificado las **funciones de pertenencia** PEQUEÑO y GRANDE del conjunto borroso E_d , siendo estas más exageradas debido a que las distancias que tiene que recorrer el robot a los puntos de destino son muy pequeñas. Las funciones modificadas y las reglas son las siguientes:



1. If (E_d is PEQUEÑA) then (V is PEQUEÑA) (1)
2. If (E_d is MEDIA) then (V is PEQUEÑA) (1)
3. If (E_d is GRANDE) then (V is PEQUEÑA) (1)
4. If (E_{θ} is NEGATIVO) then (W is NEGATIVA) (1)
5. If (E_{θ} is POSITIVO) then (W is POSITIVA) (1)
6. If (E_{d_obs} is PEQUEÑA) then (W is POSITIVA) (1)

A continuación, se muestran **varias trayectorias** seguidas por el robot en las que está obligado a esquivar un obstáculo:





Durante la realización de toda la práctica es importante destacar dos cosas:

1. Se ha buscado antes la **precisión** que la velocidad.
2. Resulta muy útil la **visualización de reglas** en la herramienta fuzzy para poder conocer mejor el comportamiento del controlador.