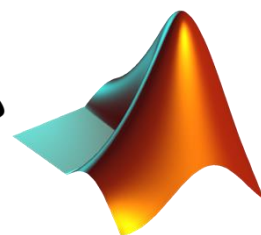
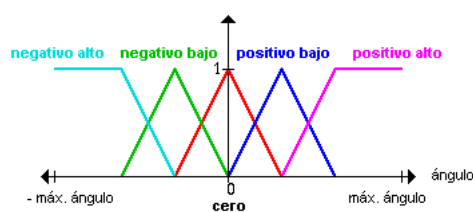


Sistemas de control Inteligente

Ejercicios de Control Borroso

Juan José Córdoba Zamora
Juan Casado Ballesteros



Universidad
de Alcalá

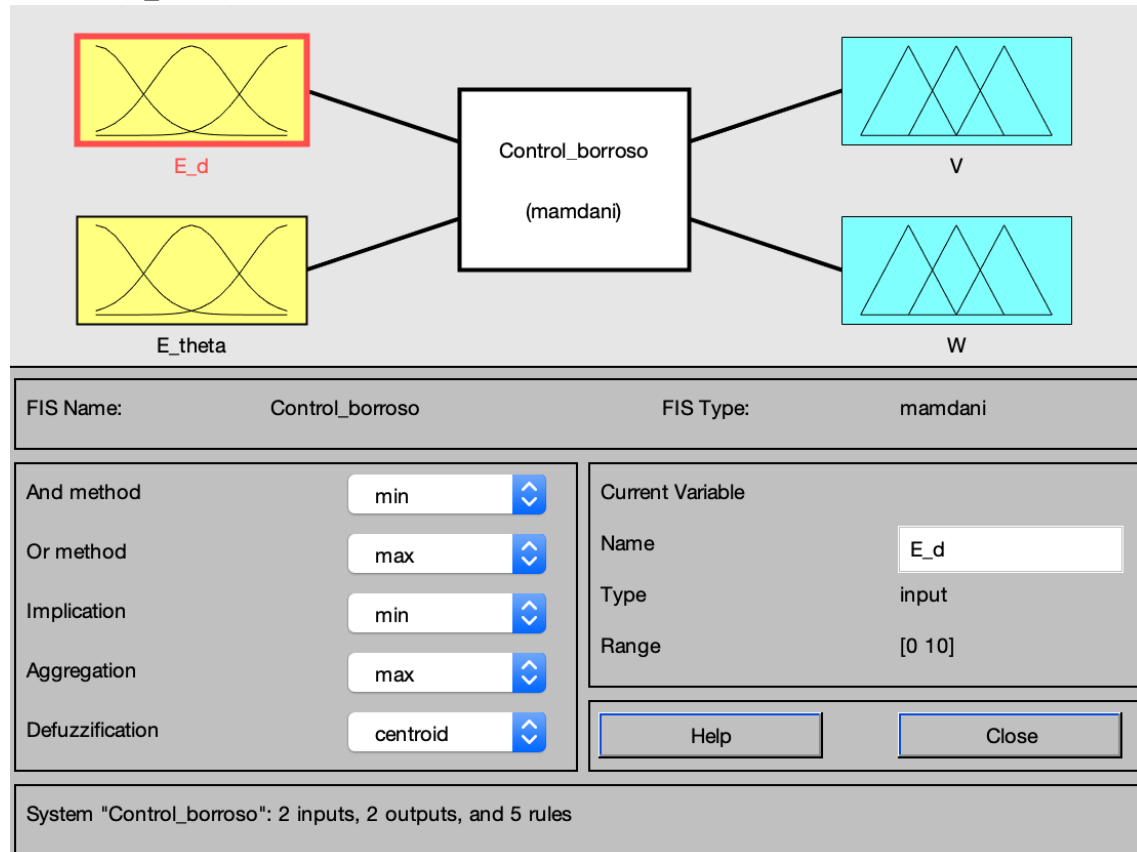
ÍNDICE

IR A UN PUNTO SIN OBSTÁCULOS	3
CONTROL BORROSO BASE	4
CONTROL BORROSO CON GAUSSIANAS	5
CONTROL BORROSO RÁPIDO	6
RESULTADOS OBTENIDOS	7
SEGUIR LA TRAYECTORIA SIN OBSTÁCULOS	10
CONTROLADOR BORROSO BASE	10
CONTROLADOR BORROSO CON GAUSSIANAS	10
CONTROLADOR BORROSO RÁPIDO	10
IR A UN PUNTO CON OBSTÁCULO	12
CONTROL BORROSO BASE	14
CONTROL BORROSO CON GAUSSIANAS	15
CONTROL BORROSO RÁPIDO	16
RESULTADOS OBTENIDOS	17
SEGUIR LA TRAYECTORIA CON OBSTÁCULO	20
CONTROLADOR CON GAUSSIANAS	20
CONTROLADOR RÁPIDO	20
CONTROLADOR BASE	20

Ir a un punto sin obstáculos

Hemos creado un total de tres controladores distintos. En cada uno hemos utilizado diferentes aproximaciones para llegar a soluciones similares que exploraremos a continuación.

En ambos controladores tendremos como variables de entrada el error entre nuestra posición actual y el objetivo (E_d) y el error entre nuestra orientación y la del objetivo respecto de nosotros (E_{θ}).



Utilizaremos mínimo como el operador T-norma y máximo como operador S-norma siendo el centroid el desborrosificador.

Los dominios utilizados para cada variable son los indicados en la memoria de la práctica.

- E_d : [0, 10]
- E_{θ} : $[-\pi, \pi]$ Redondeados a 3.15 para evitar warnings al salirnos del dominio.
- V : [0, 2]
- W : [-1, 1]

Control borroso base

Utilizamos triángulos para modelar los conjuntos borrosos.

Las reglas que dominan el controlador son simples pues el problema también lo es.

Logramos muy buenos resultados logrando llegar a la mayoría de los puntos objetivo. Los únicos a los que no se logra llegar son los que se sitúan justo detrás del robot. Para llegar a estos puntos sería necesario detectar ese caso concreto y hacer girar al robot sobre sí mismo estableciendo una velocidad lineal muy baja, casi 0. Para lograrlo hubiera sido necesario ampliar el dominio de V para que incluyera valores negativos lo que nos hubiera permitido ubicar un triángulo con centro en 0 cuyo resultado fuera una velocidad lineal nula o próxima a la nula.

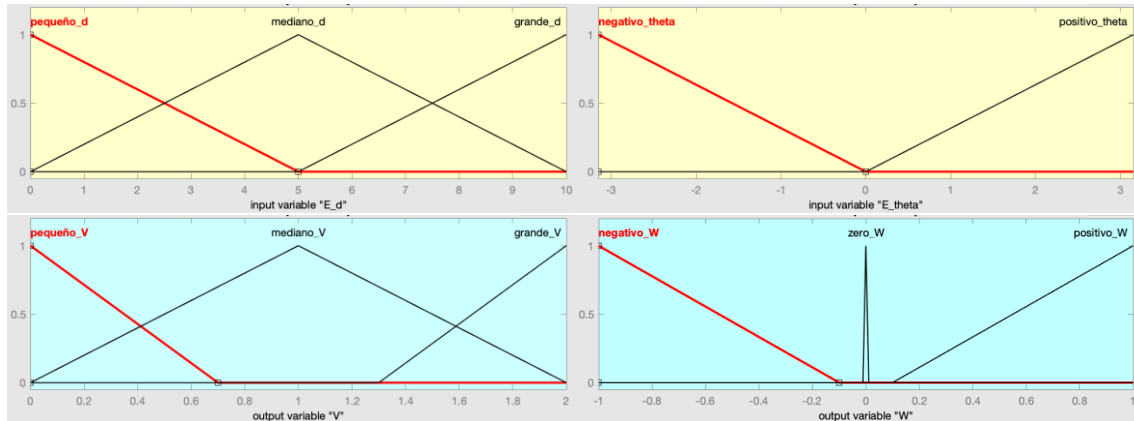


Tabla de reglas para la salida V. La velocidad lineal de salida depende de E_d, cuando esta sea mayor iremos más rápido de forma proporcional.

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	pequeño_V	mediano_V	grande_V
positivo_theta	pequeño_V	mediano_V	grande_V

Tabla de reglas para la salida W. La velocidad angular de salida depende de E_theta, cuando esta sea positiva debemos aplicar una velocidad angular positiva y viceversa.

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	negativo_W	negativo_W	negativo_W
positivo_theta	positivo_W	positivo_W	positivo_W

Adicionalmente en todo momento el conjunto de salida zero_W estará siempre activo, es decir, para cualquier valor de E_d este conjunto se activará lo cual reduce la velocidad angular haciendo que giremos un poco menos, lo suficiente como para no oscilar al seguir la trayectoria.

Para ajustar dichos conjuntos primeros establecimos una velocidad lineal fija baja de modo que probamos las reglas de W. Luego ajustamos las reglas de V y finalmente volvimos a ajustar las de W momento en el que nos dimos cuenta de la necesidad de este conjunto zero_W. Adicionalmente este conjunto evita los warnings cuando E_theta es cero ya que no habría ningún valor sobre el que el desborrosificador de W pudiera actuar.

1. If (E_d is pequeño_d) then (V is pequeño_V)(W is zero_W) (1)
2. If (E_d is mediano_d) then (V is mediano_V)(W is zero_W) (1)
3. If (E_d is grande_d) then (V is grande_V)(W is zero_W) (1)
4. If (E_theta is negativo_theta) then (W is negativo_W) (1)
5. If (E_theta is positivo_theta) then (W is positivo_W) (1)

Control borroso con gaussianas

Basándonos en lo aprendido en el controlador anterior decidimos crear este utilizando ahora conjuntos de salida y de entrada modelados mediante gaussianas.

El principal beneficio de utilizar gaussianas y además hacer que se crucen en la variable de salida **W** fue para intentar realizar giros más abiertos y que cuánto más negativo o más positivo sea el error de theta más girará al hacer que las gaussianas no terminasen en el eje $x=0$ (o incluso antes del 0) como en el controlador anterior, sino que afectará también y por lo tanto tuviera mayor área.

En cuanto a la variable **V** con la variable muy_pequeño, se ha incluido para que al girar si la velocidad es muy pequeña la circunferencia resultado es todavía mayor, por lo que esto nos ayuda a todavía realizar giros más abiertos.

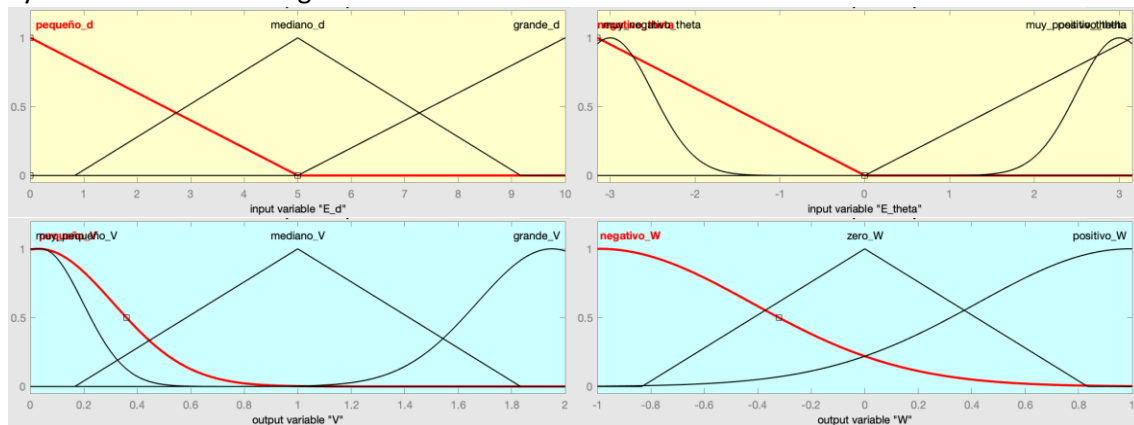


Tabla de reglas para la salida V. Este controlador se basa en realizar giros mayores pero sin aumentar demasiado la complejidad del controlador, por lo que las reglas del controlador anterior eran válidas, pero al aumentar las variables del conjunto de salida **V** pudimos realizar giros más grandes disminuyendo la velocidad (se explicará mas adelante).

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	pequeño_V	mediano_V	grande_V
positivo_theta	pequeño_V	mediano_V	grande_V

Salida W depende exclusivamente de E_theta.

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	negativo_W	negativo_W	negativo_W
positivo_theta	positivo_W	positivo_W	positivo_W

Añadimos las siguientes reglas para girar más rápido cuando el E_theta es demasiado grande, por lo que si ponemos la velocidad a muy pequeña, el giro todavía es mayor.

Reglas 6 y 7.

E_theta	
muy_negativo_theta	muy_pequeño_V
Muy_positivo_theta	muy_pequeño_V

1. If (E_theta is negativo_theta) then (W is negativo_W) (1)
2. If (E_theta is positivo_theta) then (W is positivo_W) (1)
3. If (E_d is pequeño_d) then (V is pequeño_V) (1)
4. If (E_d is mediano_d) then (V is mediano_V) (1)
5. If (E_d is grande_d) then (V is grande_V) (1)
6. If (E_theta is muy_negativo_theta) then (V is muy_pequeño_V)(W is negativo_W) (1)
7. If (E_theta is muy_positivo_theta) then (V is muy_pequeño_V)(W is positivo_W) (1)

Control borroso rápido

En este último controlador decidimos centrarnos en obtener una mayor rapidez para alcanzar el objetivo. Esto lo logramos aumentando también la complejidad de controlador tal y como explicaremos a continuación.

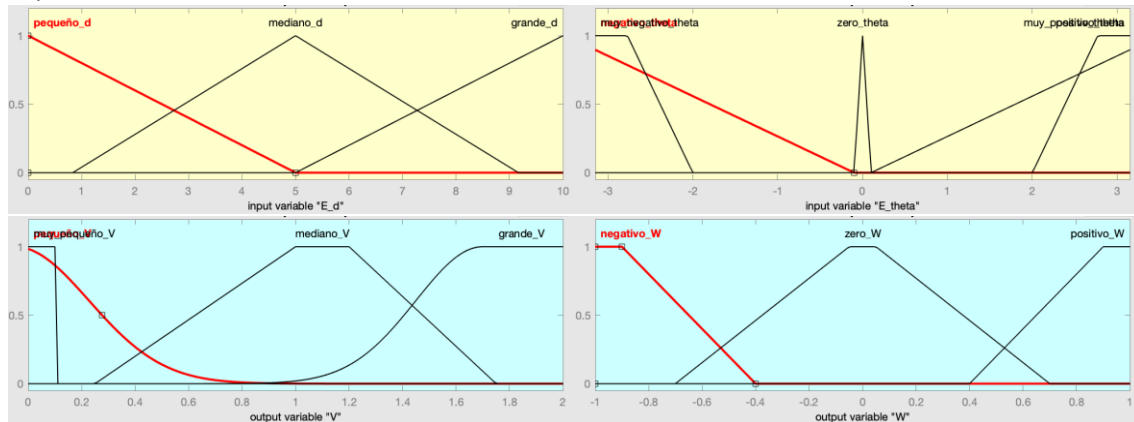


Tabla base para la salida V. Al igual que en el primer controlado produciremos velocidades proporcionales al error. La diferencia en las velocidades de salida entre ambos reside en la forma de los conjuntos borros utilizados. La velocidad media está desplazada hacia valores más grandes de lo normal mientras que las velocidades grandes y pequeñas están saturadas hacia sus extremos respectivo modeladas mediante gaussianas.

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	pequeño_V	mediano_V	grande_V
positivo_theta	pequeño_V	mediano_V	grande_V

Del mismo modo actuamos sobre la salida W que depende principalmente de E_theta.

E_theta/E_d	pequeño_d	mediano_d	grande_d
negativo_theta	negativo_W	negativo_W	negativo_W
positivo_theta	positivo_W	positivo_W	positivo_W

Añadimos las siguientes reglas para girar más rápido cuando el E_theta es demasiado grande, estas reglas tratan de compensar que tendemos a ir más rápido de lo que deberíamos.

Reglas 6 y 7.

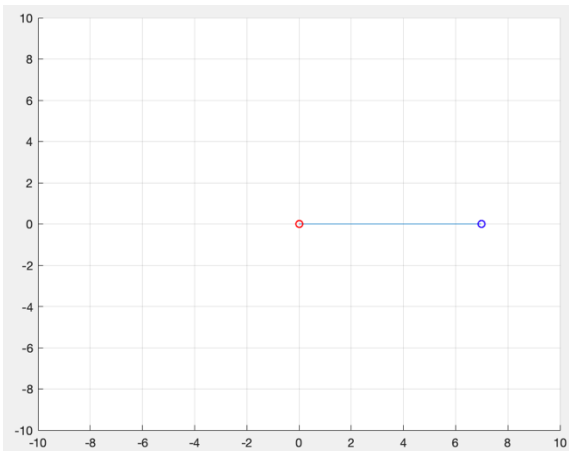
E_theta	
muy_negativo_theta	muy_pequeño_V
Muy_positivo_theta	muy_pequeño_V

También tratamos de detectar cuando E_theta es zero para proporcionar un incremento momentáneo de velocidad lineal desplazando las salidas de V al siguiente valor, es decir, si la velocidad debería de ser baja iremos a velocidad media y si debería de ser media la pondremos alta manteniendo un W de cero. Reglas 8, 9.

1. If (E_theta is negativo_theta) then (W is negativo_W) (1)
2. If (E_theta is positivo_theta) then (W is positivo_W) (1)
3. If (E_d is pequeño_d) then (V is pequeño_V) (1)
4. If (E_d is mediano_d) then (V is mediano_V) (1)
5. If (E_d is grande_d) then (V is grande_V) (1)
6. If (E_theta is muy_negativo_theta) then (V is muy_pequeño_V)(W is negativo_W) (1)
7. If (E_theta is muy_positivo_theta) then (V is muy_pequeño_V)(W is positivo_W) (1)
8. If (E_d is pequeño_d) and (E_theta is zero_theta) then (V is mediano_V)(W is zero_W) (1)
9. If (E_d is mediano_d) and (E_theta is zero_theta) then (V is grande_V)(W is zero_W) (1)

Resultados obtenidos

Los resultados obtenidos no difieren demasiado al intentar ir hacia un punto, solo varían en el tiempo que se ha necesitado para llegar al objetivo. Las diferencias se hacen más notables al intentar seguir la trayectoria.

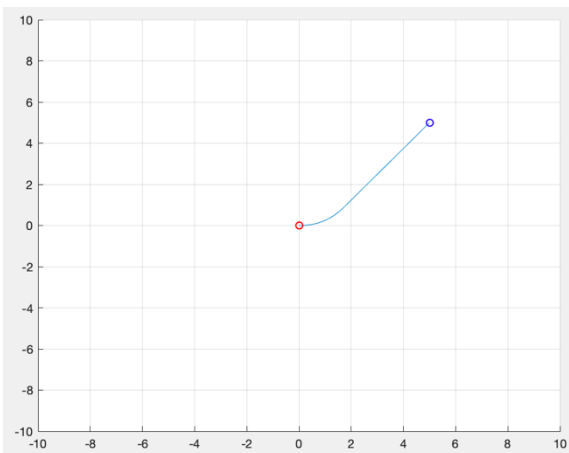


Ir en línea recta [7, 0]

Dibujaremos con un círculo color rojo la posición inicial del robot y con un círculo azul el punto objetivo.

Intervalos de tiempo en llegar al objetivo:

- Controlador base: 87
- Controlador con gaussianas: 112
- Controlador rápido: 67

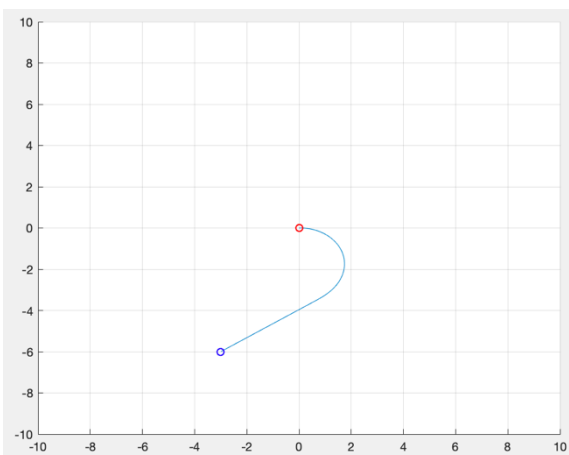


Punto en diagonal [5, 5]

El robot se dirige hacia el punto haciendo primero un giro mientras avanza hasta centrarse con respecto a él y luego siguiendo en línea recta.

Intervalos de tiempo en llegar al objetivo:

- Controlador base: 89
- Controlador con gaussianas: 117
- Controlador rápido: 80



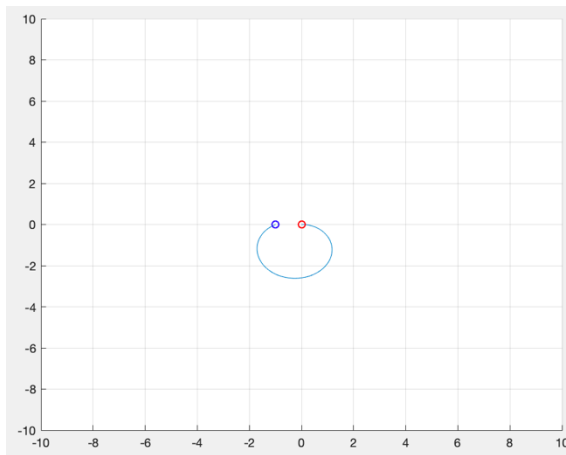
Punto en el cuadrante negativo [-3, -6]

De modo similar al anterior primero nos orientamos hacia el punto mientras avanzamos y luego vamos rectos.

La curva inicial es más abierta.

Intervalos de tiempo en llegar al objetivo:

- Controlador base: 108
- Controlador con gaussianas: 150
- Controlador rápido: 97



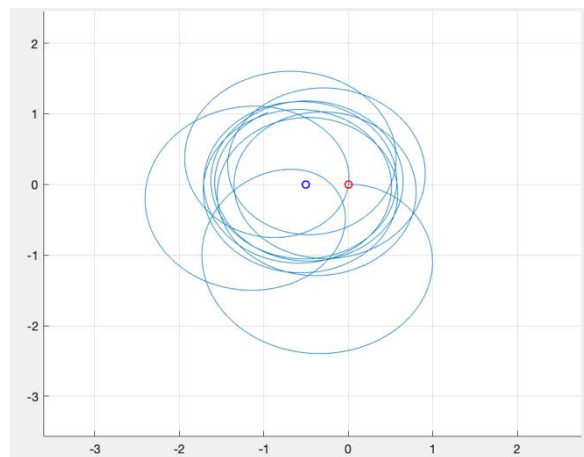
Punto detrás del robot $[-1, 0]$

Con puntos más cercanos el controlador no llega a producir una W suficientemente grande, se podría aumentar el dominio de esta para lograrlo.

Intervalos de tiempo en llegar al objetivo:

- Controlador base: 104
- Controlador con gaussianas: 173
- Controlador rápido: 81

Es importante destacar que si el punto final está detrás y todavía más cerca del punto inicial que en la imagen de arriba (entre $[-1,0]$ y $[0,0]$) el **controlador base** no es capaz de realizar la trayectoria, como se ha comentado anteriormente.

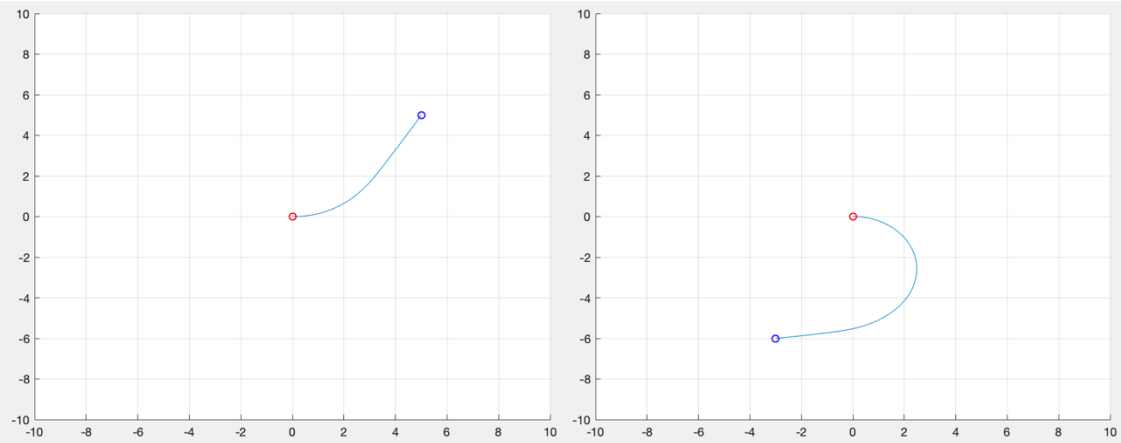


Las capturas mostradas son del controlador base. Se puede ver como en todos los casos el controlador rápido llega al punto en menor tiempo mientras que el controlador con gaussianas tarda más. Esto se debe a que el controlador con gaussianas tiene una velocidad lineal base menor además de que realiza las curvas más abiertas. El controlador rápido intenta hacer todo lo contrario, girar más en las curvas e ir más rápido en las rectas.

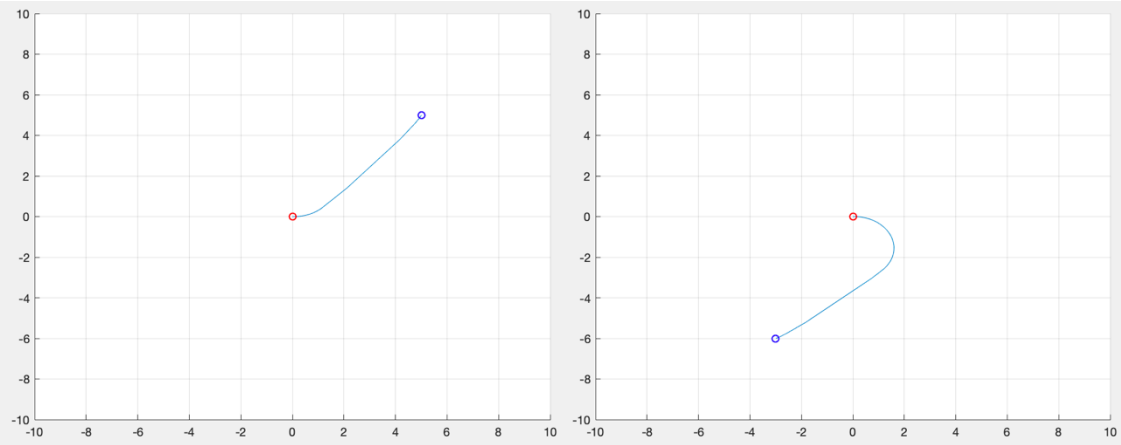
Los tres controladores logran llegar al punto objetivo.

Mostraremos ahora capturas ilustrativas de los otros controladores para ver las diferencias explicadas sobre la trayectoria que describen.

Controlador con gaussianas: las curvas generadas son más abiertas.



Controlador rápido: las curvas generadas son más cerradas.



Seguir la trayectoria sin obstáculos

Los controladores que hemos mostrado habían sido ya ajustados para seguir la trayectoria. Según estábamos realizando la práctica fue necesario realizar pequeñas modificaciones en el tamaño de los conjuntos de entrada y salida para lograr seguir la trayectoria correctamente. Estas modificaciones fueron mínimas y no requirieron añadir reglas nuevas, solo ajustar el tamaño de los conjuntos.

Mostraremos el punto donde se inicia el robot en rojo y donde se inicia la trayectoria en amarillo, el controlador deberá llegar primero al punto amarillo y luego seguir la trayectoria.

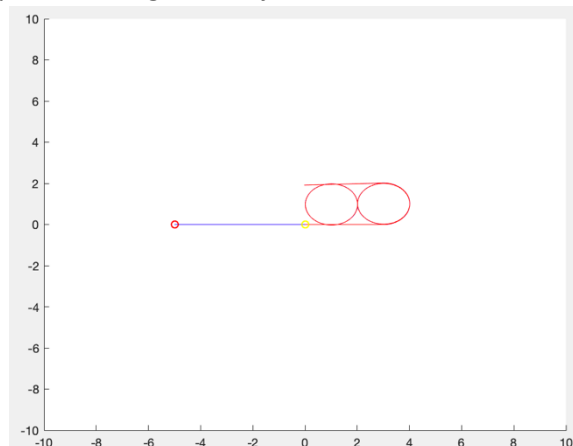
La trayectoria del robot se pinta en azul y ya trayectoria a seguir en rojo.

Controlador borroso base

Como vemos este controlador sigue la trayectoria yendo justo encima de ella sin desviarse.

En todo momento el robot está por detrás de los puntos a seguir.

Intervalos de tiempo: 876

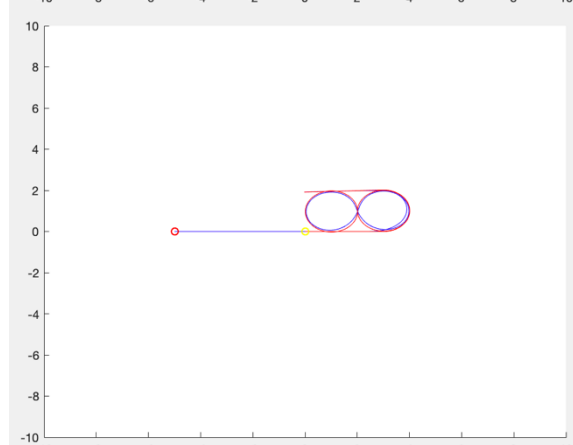


Controlador borroso con gaussianas

Este controlador tiene pequeños problemas para realizar los giros ya que intenta hacer las curvas más abiertas. Si la trayectoria a seguir hubiera tenido curvas más cerradas no hubiera podido hacerlo.

Debido a su complejidad ajustarlo mejor no fue posible.

Intervalos de tiempo: 906

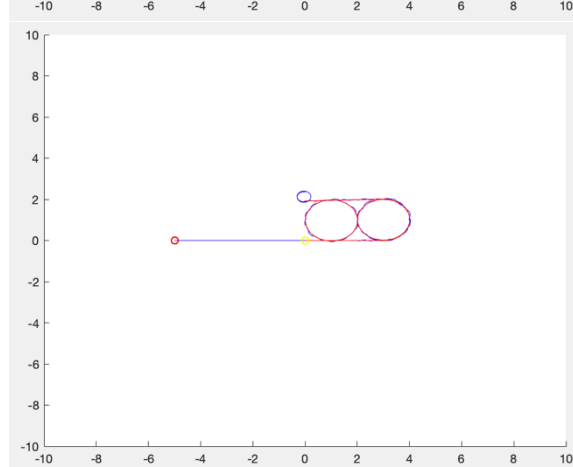


Controlador borroso rápido

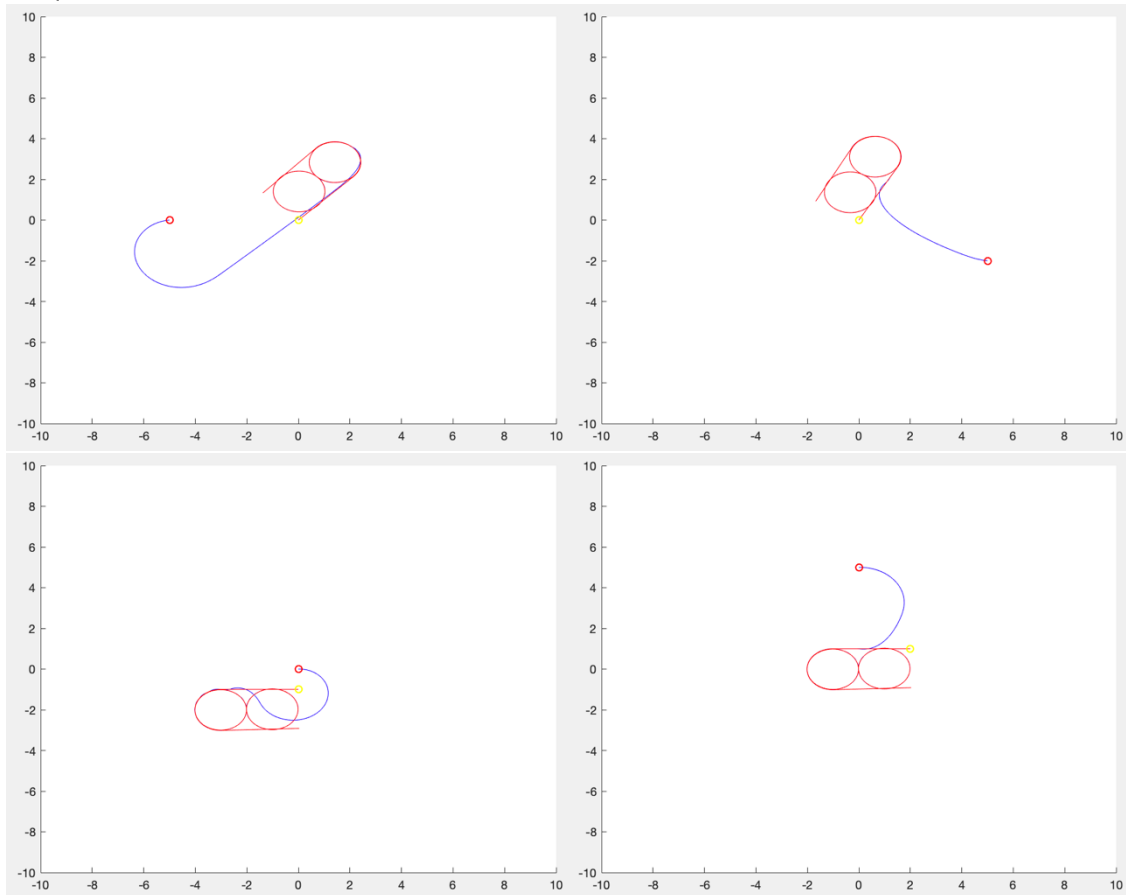
El controlador rápido va demasiado rápido como para seguir la trayectoria correctamente. Alcanza a los puntos de la trayectoria y se sitúa delante de ellos para seguirlos.

Esto tiene dos consecuencias, se producen oscilaciones en la trayectoria que él describe que está siguiendo a un punto que tiene detrás. Adicionalmente se ha tenido que eliminar la condición de parada por lo que esta sucede cuando la simulación termina lo cual le hace girar entorno al último punto de la trayectoria cuando termina.

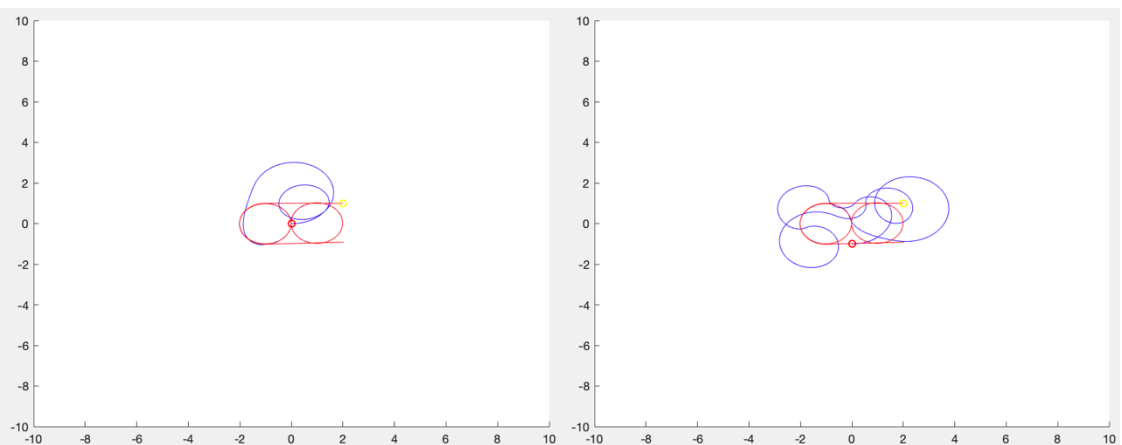
Intervalos de tiempo: 1001 esta medida no es válida ya que alcanza el punto mucho antes, no obstante, es mayor que el resto ya que debemos esperar a que termine la simulación. Realmente debería ser menor que con el controlador base.



Cuando la trayectoria está en otra posición los controladores también son capaces de adaptarse a ella. Se van aproximando al último punto que se haya generado y comienzan a seguirla una vez que lo alcanzan.



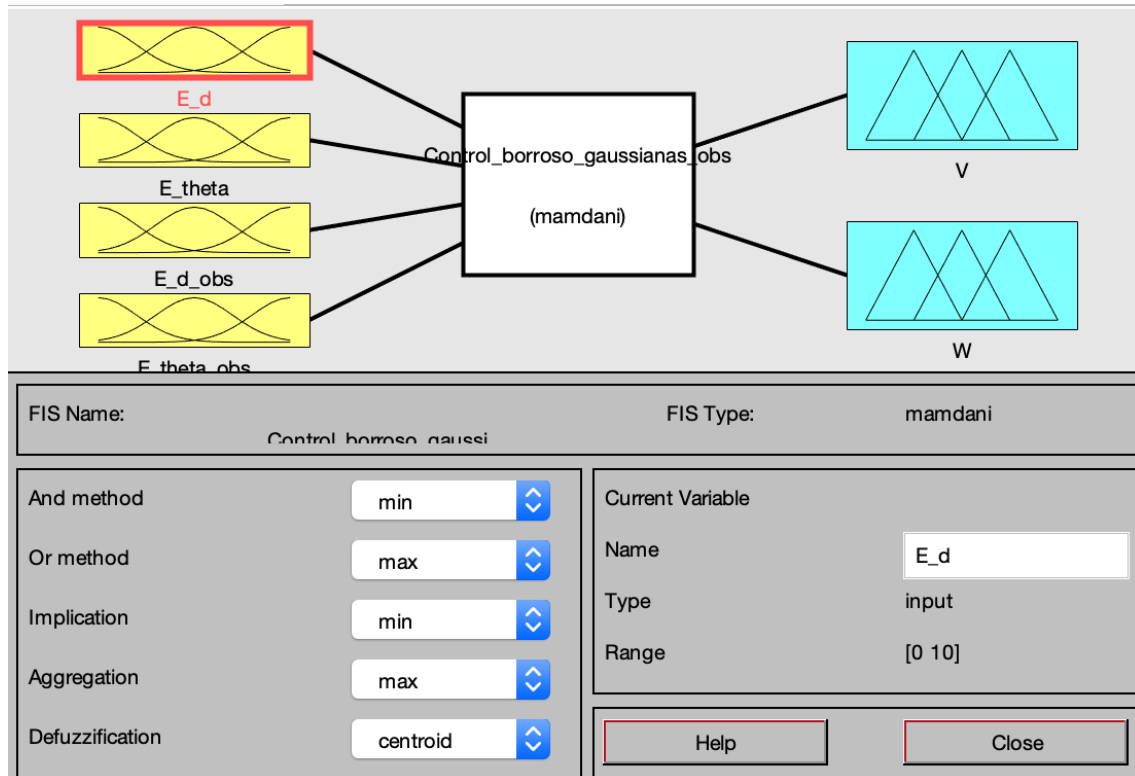
No obstante, cuando la trayectoria descrita por el robot para incorporarse a la trayectoria rompe con la evolución que harán los puntos en el futuro no logrará realizar dicha incorporación y por tanto no seguirá la trayectoria. Aunque seguirá intentando alcanzar el último punto generado a toda costa.



En ambas situaciones el robot acaba logrando unirse a la trayectoria, aunque lo hace demasiado tarde como para considerar que la está siguiendo correctamente.

Ir a un punto con obstáculo

Para la realización de este apartado se han utilizado los 3 controladores anteriormente descritos, aunque se han realizado modificaciones tanto en las reglas, como en las entradas del controlador creando nuevas. En todas se han añadido las variables de entrada E_d_obs y E_theta_obs que indican el error de la distancia y el error del ángulo al punto que hay que esquivar respectivamente.



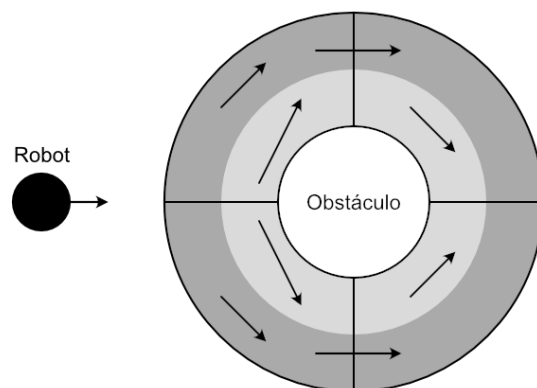
Para crear los controladores se ha partido de la idea de crear un campo de repulsión entorno al obstáculo. Cuando el robot se acerque a él el obstáculo le echará hacia los lados siendo la repulsión mayor cuanto más cerca estemos del obstáculo.

Ya que en el controlador borroso simple no tenemos distintas intensidades de velocidad angular para este en concreto el campo de repulsión será siempre el mismo sin importar la distancia al obstáculo.

Posteriormente cuando hayamos dejado atrás el objeto este nos atraerá hacia él de modo que en los bordes iremos rectos por efecto del control borroso sin que sea necesario indicarlo explícitamente.

A esta idea ha sido necesario añadir un punto de control más para que el controlador esquive el objeto justo cuando va recto hacia él. Esto lo hemos solventado detectando ese punto y desviando el robot con una W negativa.

Partiendo de las mismas configuraciones que los controladores utilizados en la parte anterior añadimos nuevas entradas para detectar el obstáculo y nuevas reglas para esquivarlo.

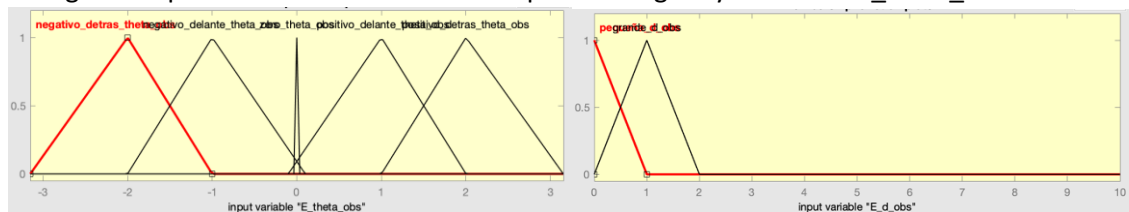


Utilizaremos los mismos conjuntos difusos para esquivar el obstáculo en los tres controladores. Dichos conjuntos buscan detectar los ocho campos delimitados en la figura anterior. Necesitaremos saber si estamos delante o detrás del obstáculo y cómo de cerca estamos con respecto a él. Adicionalmente necesitaremos detectar cuando el robot está justo alineado con el obstáculo para desviarlo hacia alguno de los 8 campos.

La variable de entrada E_d_obs tiene dos triángulos que están ubicados uno donde la distancia es muy pequeña y otro donde solo es pequeña, esto es debido a que solo nos preocupa cuando la distancia al punto es relativamente pequeña, para no afectar al robot cuando esté lejos del obstáculo y no se encuentra en la zona de repulsión descrita anteriormente.

La variable de entrada E_theta_obs se han tiene 5 triángulos, 2 para identificar si se encuentra delante, dos para identificar si se encuentra detrás.

En las reglas para identificar si se encuentra delante o detrás se identifican también si el ángulo es negativo o positivo creando los ocho campos de la figura y siendo $zero_theta_obs$ el noveno.



El dominio de las variables coincide con el de sus semejantes en el control sin obstáculos:

- E_d_obs : $[0, 10]$
- E_theta_obs : $[-\pi, \pi]$ aproximando π a 3.15 para evitar warnings.

Las reglas creadas a partir de estos conjuntos serán las mismas para los controladores rápido y gaussiano siendo ligeramente distintas para el controlador base ya que este carece de algunos de los conjuntos borrosos de salida que los otros si tienen.

Se han reutilizado en los tres controladores para poder observar los resultados que las reglas que ya existían en ellos afectaban a conjuntos borrosos idénticos de modo que las comparaciones pudieran ser más interesantes. Para ajustar el tamaño de los conjuntos se ha llegado a un compromiso entre el tamaño que mejor funcionaba en cada controlador.

Control borroso base

A las reglas anteriormente descritas para este controlador se han añadido 7 nuevas reglas. Estas reglas actuarán junto a las ya definidas y explicadas anteriormente.

Tabla de reglas para la salida V, teniendo en cuenta únicamente E_d_obs y E_theta_obs.

D / theta	negativo_detrás	negativo_delante	zero	Positivo_delante	positivo_detrás
pequeño	pequeña	pequeña	pequeña	pequeña	pequeña
grande		pequeña	pequeña	pequeña	

Cerca del obstáculo iremos siempre despacio excepto si ya lo hemos pasado y estamos lejos que dejamos a las otras reglas actuar

Tabla de reglas para la salida W, teniendo en cuenta únicamente E_d_obs y E_theta_obs.

D / theta	negativo_detrás	negativo_delante	zero	positivo_delante	positivo_detrás
pequeño	negativo	positivo	negativo	negativo	positivo
grande		positivo	negativo	negativo	

Delante del objeto lo esquivamos apartándonos de él y detrás del objeto recuperamos nuestra trayectoria anterior solo cuando estemos cerca del objeto todavía, en los otros dos casos dejaremos al resto de reglas actuar.

6. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
7. If (E_d_obs is grande_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
8. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
9. If (E_d_obs is grande_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
10. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_detrás_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
11. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_detrás_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
12. If (E_theta_obs is zero_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)

Control borroso con gaussianas

Las reglas que se aplicarán sobre este controlador son en el fondo las mismas que ya se utilizaron en el anterior. Sus diferencias residen en que tanto en este como en el controlador rápido tenemos más conjuntos de salida de modo que podemos ser un poco más creativos y ajustar mejor nuestras reglas con el fin de lograr mejores resultados.

Tabla de reglas para la salida V, teniendo en cuenta únicamente E_d_obs y E_theta_obs.

D / theta	negativo_detrás	negativo_delante	zero	Positivo_delante	positivo_detrás
pequeño	pequeña	muy pequeña	muy pequeña	muy pequeña	pequeña
grande		pequeña	muy pequeña	pequeña	

Cerca del obstáculo iremos siempre despacio, si estamos muy cerca por delante o alineados con el objeto iremos muy despacio excepto si ya lo hemos pasado y estamos lejos que dejamos a las otras reglas actuar.

Tabla de reglas para la salida W, teniendo en cuenta únicamente E_d_obs y E_theta_obs.

D / theta	negativo_detrás	negativo_delante	zero	positivo_delante	positivo_detrás
pequeño	negativo	positivo	negativo	negativo	positivo
grande		positivo	negativo	negativo	

Delante del objeto lo esquivamos apartándonos de él y detrás del objeto recuperamos nuestra trayectoria anterior solo cuando estemos cerca del objeto todavía, en los otros dos casos dejaremos al resto de reglas actuar.

8. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is muy_pequeño_V)(W is negativo_W) (1)
9. If (E_d_obs is grande_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
10. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is muy_pequeño_V)(W is positivo_W) (1)
11. If (E_d_obs is grande_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
12. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_detrás_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
13. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_detrás_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
14. If (E_theta_obs is zero_theta_obs) then (V is muy_pequeño_V)(W is negativo_W) (1)

Control borroso rápido

Este controlador comparte las mismas reglas que el anterior ya que ambos disponen de los mismos nombres de conjuntos de entrada y de salida. No son los mismos en forma, pero sí en concepto.

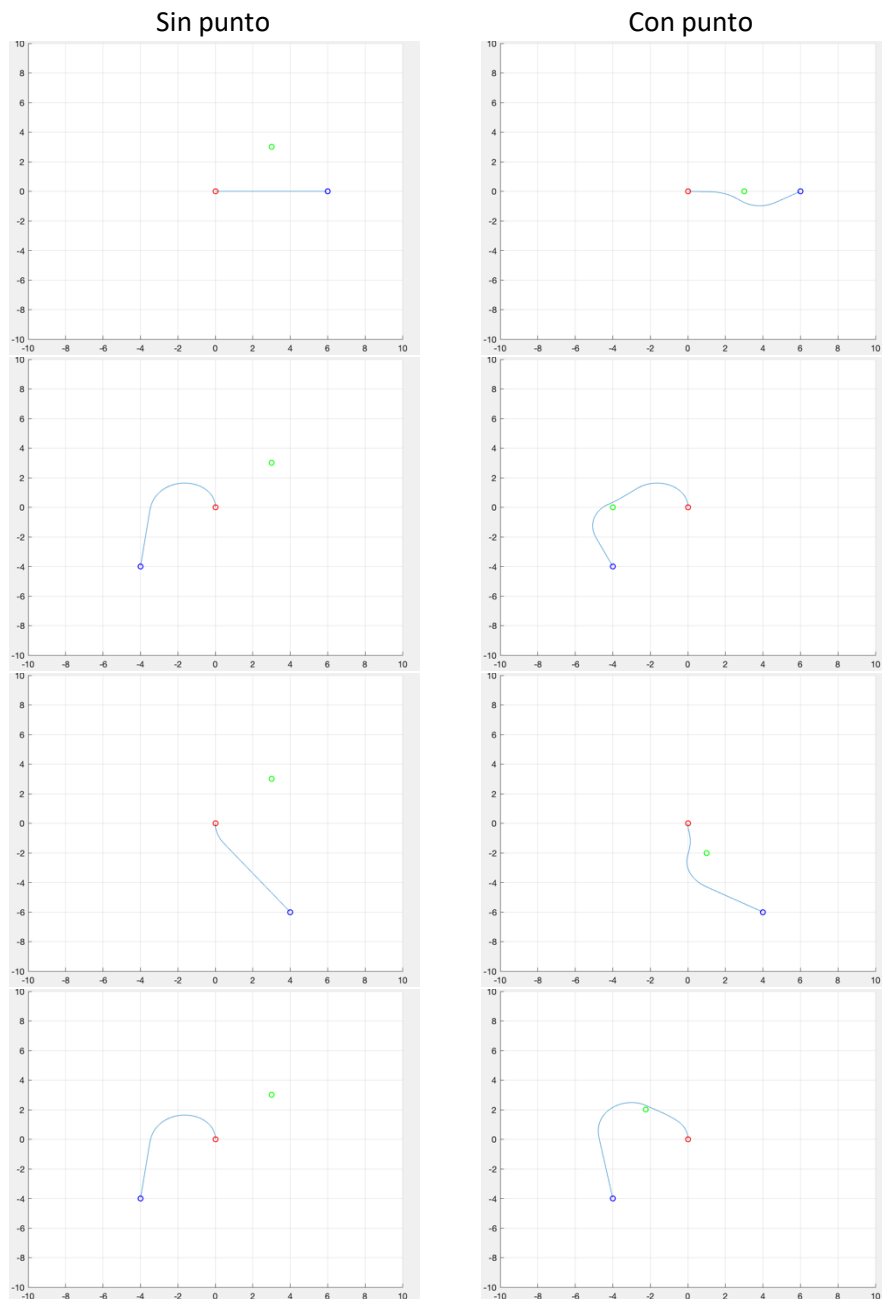
Esto nos ha permitido comprobar la facilidad y ventajas del control borroso a la hora de diseñar ya que se utilizan conceptos de tan alto nivel.

10. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is muy_pequeño_V)(W is positivo_W) (1)
11. If (E_d_obs is grande_d_obs) and (E_theta_obs is negativo_delante_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
12. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_detrás_theta_obs) then (V is pequeño_V)(W is positivo_W) (1)
13. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is negativo_detrás_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)
14. If (E_theta_obs is zero_theta_obs) then (V is muy_pequeño_V)(W is negativo_W) (1)
15. If (E_d_obs is pequeña_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is muy_pequeño_V)(W is negativo_W) (1)
16. If (E_d_obs is grande_d_obs) and (E_theta_obs is positivo_delante_theta_obs) then (V is pequeño_V)(W is negativo_W) (1)

Resultados obtenidos

Después de explicar las reglas utilizadas y como se ha hecho para poder esquivar el punto, se procede a realizar un ejemplo para poder visualizar los resultados dependiendo de la trayectoria y de dónde se encuentre el punto que tiene que esquivar el controlador, para ello se realizarán las mismas pruebas para los 3 controladores y se comentarán cuál de los 3 lo hace de la mejor forma.

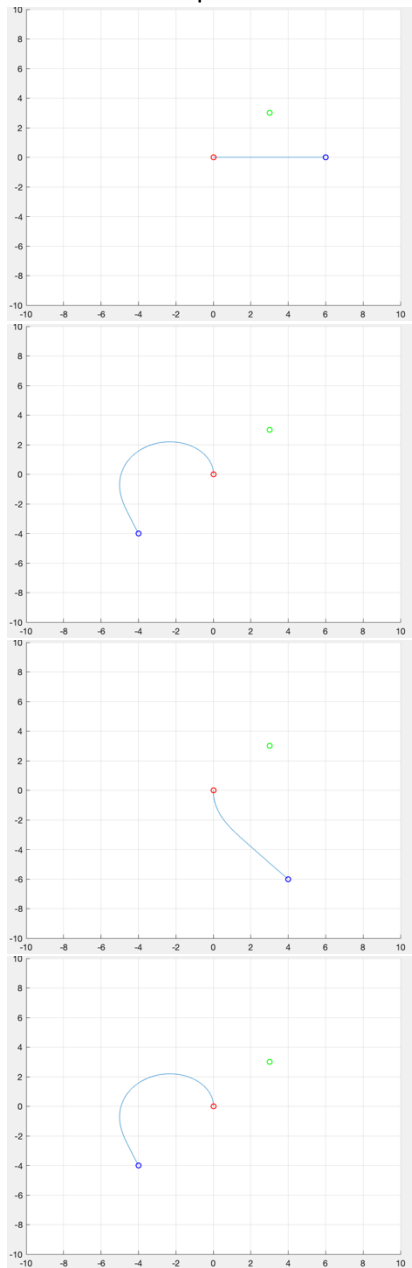
Controlador borroso base



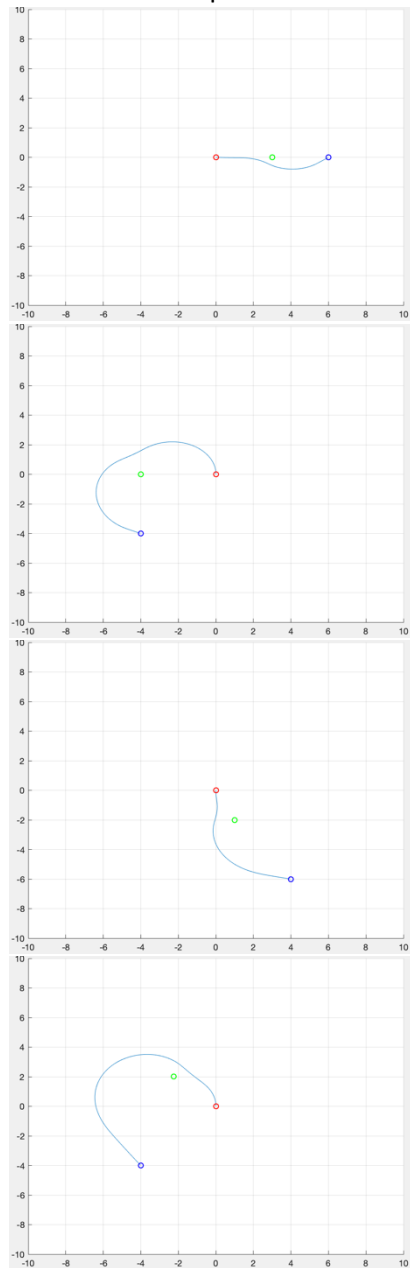
Como se puede observar este controlador responde muy bien al intentar esquivar los puntos, y en todos los casos consigue esquivar el punto de forma correcta y sin desviarse mucho de su trayectoria. Aunque en todos los casos esquiva el punto por el exterior, nunca por el interior, aunque para eso tenga que modificar más su trayectoria, esto se debe a que no puede girar lo suficiente como para hacerlo.

Controlador borroso con gaussianas

Sin punto

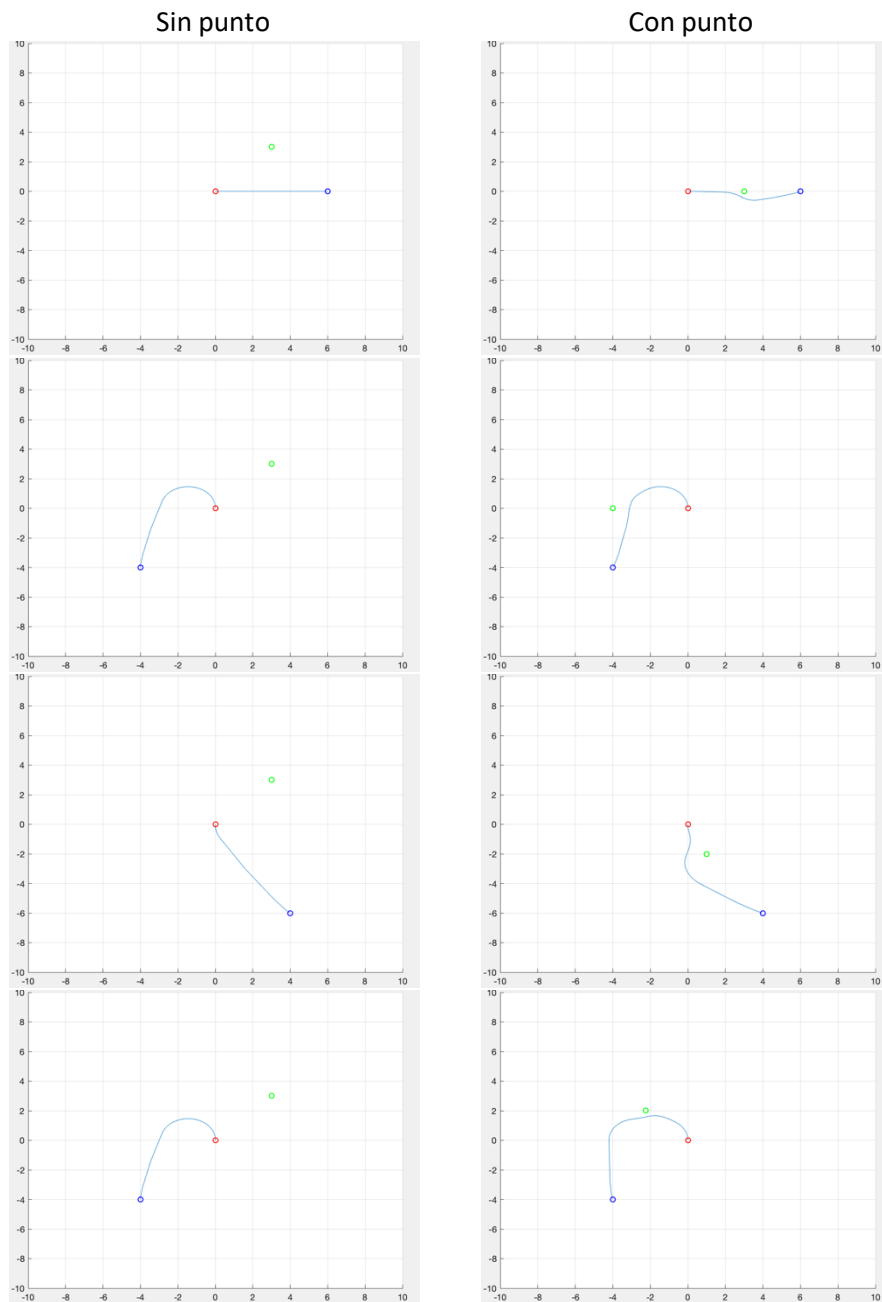


Con punto



Con este controlador se observa como efectivamente es capaz de esquivar los puntos, aunque la trayectoria que describe para esquivarlos es mucho mayor. Aunque esto parezca una desventaja, puede ser una ventaja si el punto es muy grande ya que, con el controlador anterior, aunque era capaz de esquivarlo no se separaba mucho de él. Deja una mayor distancia de seguridad y lo hace de forma controlada pues finalmente logra llegar a su destino.

Controlador borroso rápido



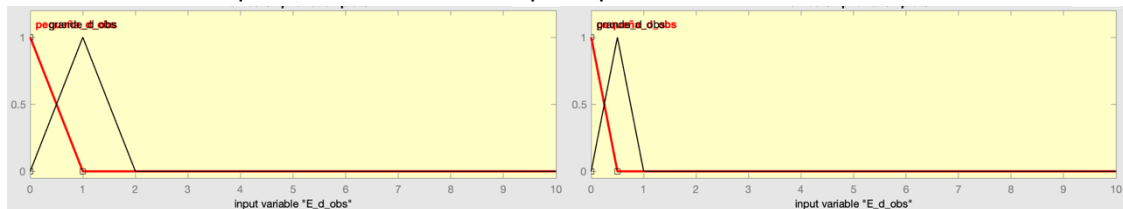
Podemos observar que en la prueba 2, la trayectoria ha sido tan buena (ha hecho poco recorrido), que el punto a penas le ha afectado pues al ir más rápido ha estado menos tiempo bajo su influencia, no como con dos anteriores controladores.

En las otras 3 pruebas se observa cómo se separa del obstáculo, pero lo hace de forma distinta que en los casos anteriores, ya que no se aleja tanto, solo lo suficiente, además los tiempos de esta prueba son menores que los de la anteriores. En la última prueba se puede ver que el controlador ha esquivado el punto por el interior ya que tiene una capacidad de giro mayor.

Seguir la trayectoria con obstáculo

Ya que la trayectoria no estaba siendo considerada en el punto anterior carecíamos de una referencia de tamaño para saber cuanto estábamos esquivando el obstáculo.

Teniendo ahora el tamaño de la trayectoria a seguir en cuenta descubrimos que en realidad estábamos esquivando los obstáculos con mucho margen. Para esquivarlos dejando una distancia de seguridad menor solo tuvimos que modificar los conjuntos borrosos en los que definíamos nuestra proximidad al obstáculo para que esta fuera menor.



Antes

Ahora

Ajustar otras distancias de seguridad solo supondría volver a modificar estos conjuntos.

Teniendo estos mismos conjuntos para los tres controladores comparamos los resultados obtenidos en varias situaciones de interés para comparar como cada controlador reacciona.

Controlador con gaussianas

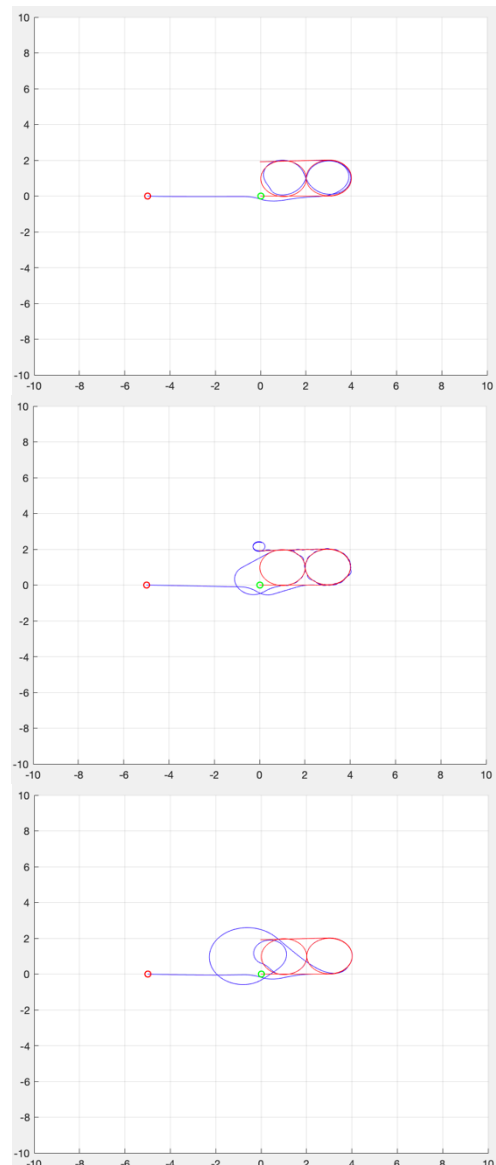
Este controlador tiene una capacidad de giro menor, es por ello por lo que cuando va recto hacia el obstáculo lo esquiva con un margen menor, posteriormente cuando se acerca a él tras la curva no es capaz de esquivarlo por el exterior como debería de modo que lo acaba haciendo por el interior lo que en este caso le favorece pues acaba haciendo menos recorrido.

Controlador rápido

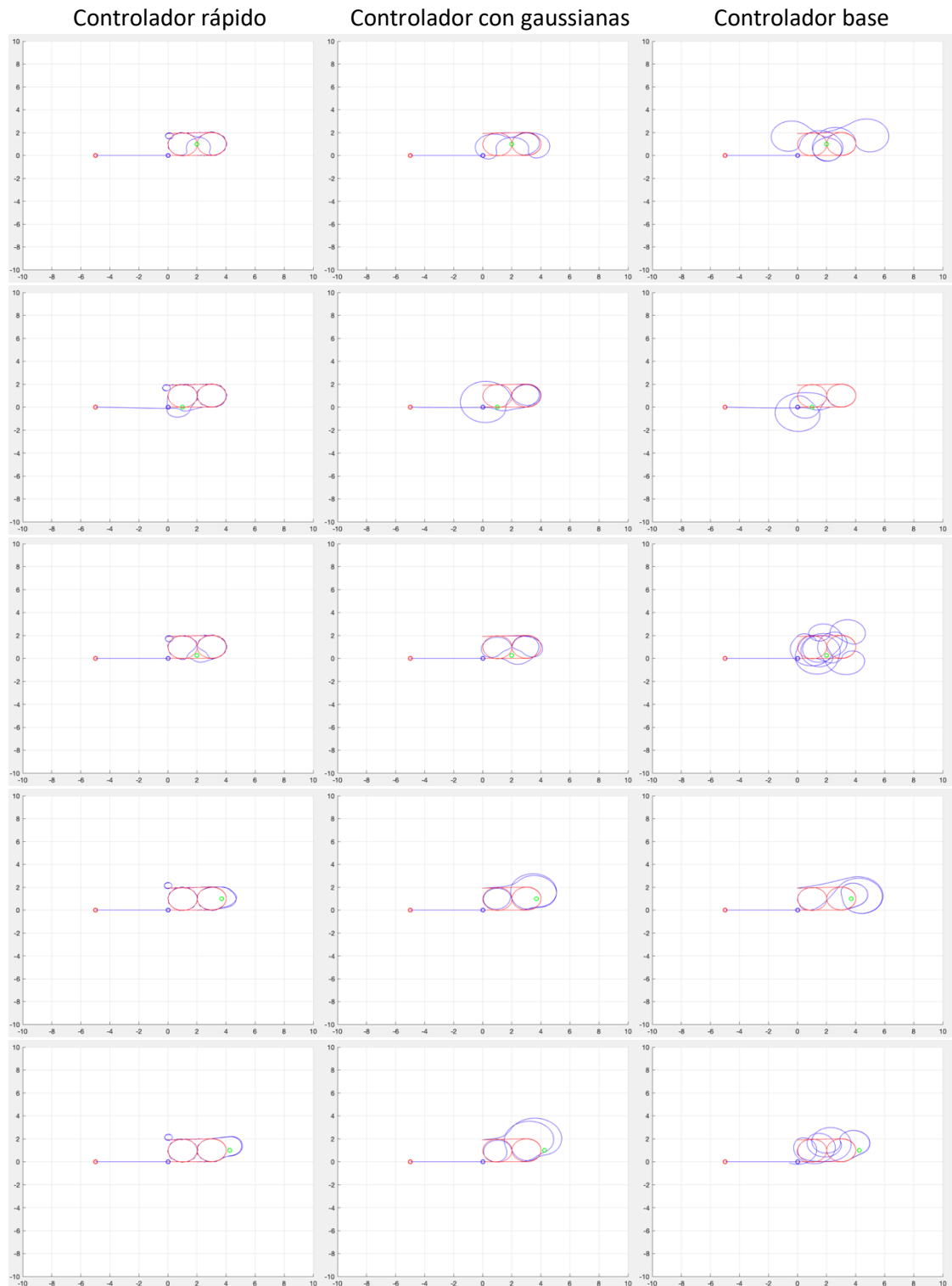
Este controlador a diferencia del anterior tiene una capacidad de giro mayor. Cuando se acerca directo al obstáculo lo esquiva dejando una distancia de seguridad considerable. Posteriormente cuando se acerca a él por segunda vez tras la curva tiene capacidad de giro suficiente como para esquivarlo por el exterior como debería. En este caso concreto eso le perjudica pues acaba recorriendo más camino.

Controlador base

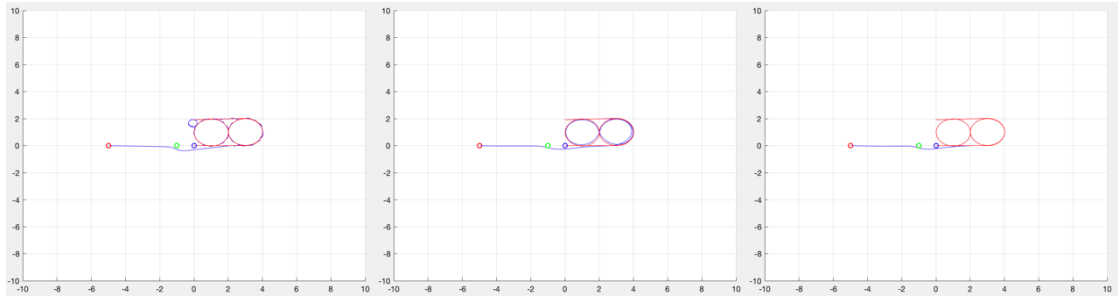
El punto realmente estaba ubicado estratégicamente para que funcionara especialmente mal con este controlador. Al esquivarlo ya que no podemos reducir más nuestra velocidad lineal ya que carecemos de esos conjuntos acabamos poniéndonos por delante de la trayectoria. Cuando el robot intenta volver a ella, pues se ha alejado de la influencia del obstáculo acaba girando rápidamente sin control hasta finalmente lograrlo.



En los siguientes ejemplos se puede ver como el controlador rápido es el que mejor se adapta a la trayectoria esquivando el obstáculo, el gaussiano lo hace, pero trazando curvas de mayor amplitud y el base no lo logra por falta de recursos.



Todos los controladores son capaces de esquivar es obstáculo y seguir la trayectoria en el caso más simple siendo el controlador rápido el que más margen deja con respecto al obstáculo pues es capaz de girar más.



Recordamos que el controlador rápido está siguiendo la trayectoria por delante de los puntos por lo que no podemos poner condición de parada por error bajo ya que los atraviesa. Es por ello también que oscila al seguir la trayectoria. El controlador gaussiano tiene problemas para hacer curvas amplias. El controlador base sigue la trayectoria sin problemas, no obstante, su simplicidad hace que no se adapte del todo bien a esquivar el obstáculo, necesita conjuntos para poder girar más y hacerlo más lento.