Sistemas de control inteligente

Práctica de laboratorio 3

Grado en Ingeniería Informática



Raúl López Llana

Óscar Pozo de Cádiz

# Control borroso (I)

En esta práctica vamos a replicar el comportamiento de una caja negra que simula el movimiento de un robot mediante el uso de control borroso.

Para eso lo primero que haremos es crear el propio controlador borroso. A este le indicaremos que va a tener 2 entradas y 2 salidas y limitamos el rango de valores en los que va a funcionar. Estos rangos se indican en el enunciado de la práctica.

Imagen que contiene Diagrama

Descripción generada automáticamente

Los rangos como se indica en el enunciado son:



Con los rangos ya definidos, nos faltan por añadir 2 cosas. Las funciones de activación y las reglas de activación.

* Comenzamos definiendo las funciones para la señal de entrada de la distancia “E\_d”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como se puede observar, hemos decidido implementar 4 funciones ya que tras hacer distintas pruebas que se verán más adelante, observamos que con 3 funciones únicamente no desempeñaba un buen funcionamiento ya que tendía a acelerar el robot más de lo deseado debido a la distancia tan alta con su objetivo.

Con esta configuración hemos conseguido que sea capaz de seguir su objetivo con una desviación aceptable.

* Pasando ahora al apartado del ángulo de entrada “E\_theta”:

Interfaz de usuario gráfica

Descripción generada automáticamente

Hemos pensado que únicamente debíamos poner 3 funciones ya que solo puede tener 3 movimientos. Seguir recto, girar a la derecha o girar a la izquierda.

Los giros hacia los lados decidimos crearlos de forma que cuanto más lejos se encuentren de la trayectoria ideal, theta igual a 0, aumentase su regla para que pueda corregir fácilmente su trayectoria.

En cuanto al movimiento de seguir recto, creímos conveniente hacer una función que ocupase muy poco ya que queremos que realice este movimiento únicamente cuando realmente ya se encuentre en la trayectoria que queremos o muy cercana a ella.

* En cuanto a la velocidad de salida de nuestro controlador borroso:

Interfaz de usuario gráfica

Descripción generada automáticamente

Hemos considerado que la mejor opción era crear 4 funciones y dar una mayor superficie a la función “grande” ya que hemos observado que una velocidad mayor a 1,5 es demasiado para nuestro robot por lo que hemos decidido reducir el rango donde aplicar la regla “MUYGRANDE”.

* Por último, el ángulo de salida “W”

Interfaz de usuario gráfica

Descripción generada automáticamente

Siguiendo las mismas consideraciones dichas previamente para “E\_theta”, aplicamos de la misma forma para el ángulo de salida del robot.

Una vez definidas las funciones de entrada y salida, nos queda ahora definir las reglas de activación.

En este apartado hemos hecho unas reglas muy sencillas y básicas.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Dado que tenemos el mismo número de entradas y de salidas para la velocidad y el ángulo, aplicamos la misma fuerza que leemos con su salida.

Por ejemplo, ante una distancia grande, aplicamos una velocidad grande. Ante una theta positiva aplicamos un ángulo W positivo.

## Resultados

Con el controlador ya creado, nos falta ver como se comporta ante unos movimientos de entrada dados.

Para esto hacemos uso del código que se nos proporciona en el enunciado de la práctica.

Texto

Descripción generada automáticamente

Tras probar varias veces el funcionamiento, observamos que la trayectoria desde el origen definido x=-4 y y=-4 es muy directo y correcto.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Sabiendo ya que el comportamiento es bueno, pasamos a que el controlador borroso siga una trayectoria que introducimos gracias al elemento que nos aporta el enunciado.

Con este vemos que el comportamiento es bueno y para que el seguimiento pueda llegar hasta el final, debemos reducir el error de parada de 0,01 a 0,001. Junto a esto debemos indicar que debe comenzar desde una posición distinta a x=0 y=0 ya que si empieza en ese punto se acabaría instantáneamente la simulación. Gracias a este cambio vemos el siguiente resultado.

Interfaz de usuario gráfica

Descripción generada automáticamente

Gracias al siguiente trozo de código podemos ver la simulación anterior.

Texto

Descripción generada automáticamente

# Control borroso (II)

En esta segunda parte de la práctica partiremos del robot previamente desarrollado y haremos que sea capaz de esquivar un objetivo.

La parte más importante de esta segunda práctica es definir cómo debemos responder ante el objetivo, ya que dependiendo del método que utilicemos la trayectoria será distinta.

Para ello, definiremos 4 estados en los que puede estar el robot respecto al obstáculo, según el ángulo que tenga respecto a este, tal y como se indica en el siguiente diagrama (los ángulos no están a escala):

Icono

Descripción generada automáticamente

El funcionamiento es el siguiente:

* Si el robot está cerca del obstáculo y éste se encuentra en la zona de “peligro”, entonces el robot girará para conseguir una trayectoria ligeramente desviada, evitando así el choque con el obstáculo.
* Si el obstáculo está en una de las zonas de “trayectoria óptima”, simplemente se moverá hacia delante. Al moverse con un ángulo respecto al obstáculo, llegará un momento en el que el obstáculo pase a encontrarse en la zona “superada”.
* Si el obstáculo pasa a estar en la zona “superada”, consideramos que el obstáculo ha sido superado, por lo que el robot continúa con la trayectoria que tendría si el obstáculo no existiese. Si al seguir esta trayectoria el obstáculo vuelve a encontrarse en la zona “peligrosa”, el robot volvería a esquivarlo.

Para llevar a cabo este comportamiento se definen estas zonas dependiendo del error de ángulo respecto al obstáculo:

Diagrama

Descripción generada automáticamente

Para que el robot tenga en cuenta al obstáculo, este debe encontrarse muy cerca:

Gráfico

Descripción generada automáticamente con confianza media

Las nuevas reglas son:



Además, se incluyen nuevas reglas para que el robot imite el comportamiento original una vez se haya superado el obstáculo:

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Con todo esto, podemos ver el resultado de la ejecución:

Gráfico

Descripción generada automáticamenteTexto

Descripción generada automáticamente

El círculo azul indica la distancia en la que el robot tiene en cuenta el obstáculo (el robot cumple la regla cerca del obstáculo).

En la figura se distinguen tres partes distintas una vez que el robot entra en el círculo:

* Al entrar, gira ligeramente hacia la izquierda, hasta que el obstáculo se encuentra en la zona ideal.
* Después, el robot continúa en una línea ascendente.
* Finalmente, el robot continúa una trayectoria similar a la que tenía antes de entrar en el círculo.