# Memoria control

Descripción del programa

Partiendo del programa Brain base del robot proporcionado por el profesorado de la asignatura, se han implementado una serie de cambios y mejoras, teniendo en mente 3 objetivos principales: lograr que el robot encuentre inicialmente la línea y sea capaz de recuperarla en caso de perderla; asegurar que el recorrido y velocidad del robot sobre la línea sean óptimos; permitir que el robot esquive obstáculos mediante el uso de sus sónares (asegurando que deje y retome la línea correctamente al encontrarse con un obstáculo).

Por un lado, se planteó la necesidad de procurar que el robot encuentre la línea inicialmente. En este sentido, aunque es probable que se puedan encontrar soluciones mejores, nuestro código se limita a avanzar hacia adelante hasta que encuentre la línea, tras lo cual comienza el acercamiento progresivo (explicado en detalle en el siguiente apartado. Por otro lado, una vez que el robot pierda la línea, irá hacia atrás. En este sentido, nos dimos cuenta inicialmente de que a veces al volver atrás estaba girado demasiado, y le costaba volver a la línea de la que es escapó. Por tanto, está implementada una marcha atrás similar al aparcar en paralelo con un coche: si se ha salido hacia la izquierda de la línea, comienza dando marcha atrás hacia la derecha, para posteriormente dar marcha atrás a la izquierda y así reorientarse para estar lo más cerca posible de estar posicionado sobre la línea, con lo que pueda reanudar su marcha sin inconvenientes.

Respecto al propio código que afecta al acercamiento a la línea y su recorrido sobre ella, inicialmente se trato de implementar los distintos Kp, Kd y Ki vistos en clase, para luego entender qué valores para los mismos serían los idóneos para lograr que el recorrido sea óptimo. A este respecto, consideramos óptimo que cuanto más lejos esté de a línea, más rápida y pronunciadamente se acerque. Cuanto más se acerca, su velocidad y giro debe ir reduciéndose, ya que no es ideal que sobrepase la línea demasiado, pues acabaría haciendo simplemente zigzags en vez de ir lo más recto posible por la línea. Tras ir probando distintas opciones, pensamos que podríamos realizar un modelo cuadrático en Kd, pues parece tener sentido generar, a partir de un par de parámetros, una parábola que modelice bien la velocidad de movimiento según la distancia a la línea. Kp se queda haciendo referencia al ángulo de giro. Para poder encontrar los parámetros, se usó el software Geogebra, en el cuál generamos una parábola y mediante los parámetros "a" y "b" (luego introducidos en el código), fuimos probando hasta encontrar una curva que nos satisficiese.

[Aquí meter foto del modelo cuadrático]

Por último, y basándonos en el sistema de seguir paredes que ya desarrollamos para la primera tarea de la asignatura, implementamos la capacidad del robot de emplear sus sónares para encontrar y esquivar obstáculos que se interpongan en su camino. En este sentido, dividimos en dos casos posibles: que el obstáculo esté más cerca del lado izquierdo del robot, o del lado derecho. Si se da el primer caso, el robot tratará de virar hacia la derecha, siguiendo a su izquierda el obstáculo cual pared a recorrer, y bordeándola hasta que encuentre de nuevo la línea. Entonces, el robot olvidará su tarea de seguir el obstáculo, y retomará la línea. Si se da el segundo caso comentado el resultado es análogo. Se añadió además un limitador de la velocidad máxima para asegurar que el robot no choque con los obstáculos al virar para intentar rodearlo.

Ejemplos ejecutados

Se muestran a continuación los tests con los que se ha probado el robot, incluyendo la traza en rojo del recorrido del mismo sobre el mapa.

[Incluir los mapas]

[Mapa del error línea perpendicular]

Como se puede observar, en algunos casos no se logra retomar la línea tras rodear un obstáculo debido al simulador, por el tema de que al encontrarse una línea perpendicular delante no encuentra correctamente la distancia del centro, y por lo tanto no avisa al robot de que deje de seguir el obstáculo y vuelva a reencontrarse con la línea.

[Mapa de Hockenheim]

Por último, el robot recorre una recreación de Hockenheimring, circuito de Formula 1 ubicado en las cercanías de Hockenheim, Alemania.