

## **Siguiendo una trayectoria con odometría.**

En esta práctica partimos de los resultados de la anterior en la que logramos medir con precisión y dibujar en la pantalla la trayectoria que seguía nuestro robot al moverse en base a una tarea que controlaba los motores haciendo que siguiera una línea negra en el suelo.

El resultado a obtener consistía en describir una trayectoria con forma de cuadrado. Para obtener este resultado decidimos eliminar la tarea en la que nos movíamos en base a otros sensores para ahora hacer en base a los resultados obtenidos de la odometría.

### **Planteamiento.**

Al iniciar suponemos que el robot está en la posición 0,0 respecto a un sistema de coordenadas cartesiano bidimensional con una distancia recorrida de 0m y un ángulo de 0 rad. La forma de mover el robot será por tanto aplicar un control PID donde la posición en la que estamos es la que nos da la odometría respecto de la posición inicial y la posición a la que queremos ir son las esquinas del cuadrado respecto a la posición de inicio del robot.

Situaremos una de las esquinas en la posición de inicio del robot con una arista justo en frente de él, la siguiente esquina se encontrará a dos metros de la inicial siguiendo la arista mencionada. La tercera esquina estará a otros dos metros a la izquierda del robot y así hasta completar el cuadrado.

Distinguimos dos partes principales en el comportamiento del robot, avanzar recto y girar por lo que decidimos realizar una máquina de estados que cambiara ligeramente el comportamiento del robot para que se moviera de la forma deseada.

Existirá un comportamiento base el cual será un control PID aplicado sobre la distancia recorrida por el robot y otro sobre su ángulo de giro. En las rectas el ángulo de giro será cero con respecto al que tenía al iniciar la recta. En los giros la distancia a recorrer será cero respecto a la ya recorrida. Las constantes  $k_p$ ,  $k_i$ ,  $k_d$  del control respecto del ángulo de giro serán distintas para obtener un mejor rendimiento, queremos que al girar lo haga de forma suave y sin rebotes sobre el ángulo de giro consigna, pero por el contrario a la hora de avanzar recto queremos un movimiento que responda de forma más rápida a las desviaciones sobre la consigna; esto lo controlamos con la máquina de estados.

A parte de las acciones ya mencionadas en las que la máquina de estados ya es útil esta nos servirá también para reconocer cuando un estado ha finalizado y por tanto cuando cambiar a otro estado. En este caso solo existen dos estados luego reconocer a cuál cambiar es trivial.

### **Ejecución.**

Realizamos dos controles PID uno sobre la distancia recorrida y otro sobre el ángulo de giro. El resultado del de la distancia recorrida de aplicaría como velocidad base para girar las ruedas en una misma dirección y el del giro para girarlas en direcciones opuestas.

Para realizar la máquina de estados creamos dos arrays, uno que nos daría los giros y otro que daría las posiciones. Cambiar de estado supondría leer en la siguiente posición del array. Una variable booleana nos diría si el estado es de giro o de avance, con esto aplicamos unas constantes de control sobre el PID del ángulo u otras.

### **Pruebas.**

Para probar el código planteamos un método iterativo en el que primero hicimos al robot mantener un ángulo concreto, aunque lo hiciéramos girar sobre su eje este mantendría siempre el ángulo de inicio.

Posteriormente lo hicimos avanzar 2m y tras lograrlo dos metros yendo recto. Por último, le hicimos girar 90 grados para lo cual debimos pasarlos a radianes pues es en la unidad que el robot mide cuanto ha girado.

En cada iteración fuimos ajustando el PID hasta lograr un comportamiento adecuado. Lo que más nos costó fue ajustar las constantes necesarias para que se moviera recto pues no nos dimos cuenta de que estas deberían de ser distintas a las de girar. Una vez que caímos en esto cambiarlo fue sencillo pues la máquina de estados organizaba muy bien el código.

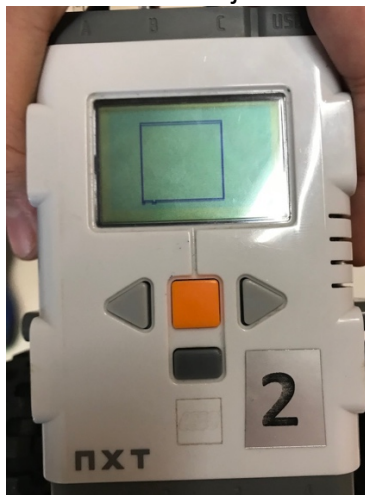
También decidimos reiniciar el acumulador del control integral a cada cambio de estado pues al hacerlo obteníamos mejores resultados debido a que para la siguiente acción no debería influir el error acumulado en la anterior.

Finalmente logramos que el robot realizara el cuadrado de forma perfecta a la vez que lo dibujaba en la pantalla.

Antes de los últimos ajustes:



Tras los últimos ajustes:



**Vídeo del robot en funcionamiento:** <https://youtu.be/ZgA75LiJO-4>