

# Práctica 2

Por Rafael Lillo López y Pedro Sánchez Machuca

## EJERCICIO 1A

En esta práctica, como primer ejercicio teníamos que comparar los resultados de hacer que los motores de las ruedas del robot lego realicen giros de  $860^\circ$ , controlándolo con un bucle abierto o con un bucle cerrado.

Comenzando por el bucle abierto, simplemente teníamos que averiguar mediante prueba y error cual era el valor de velocidad del motor que se aproximase más a estos  $860^\circ$  que se piden. En el enunciado se aclara que existe un margen de error de unos  $10^\circ$ , pero en el tiempo que se pide alcanzar esa rotación ( 2 segundos ) es totalmente imposible llegar a una cifra lo suficientemente cercana al objetivo.

Salvo por este pequeño inconveniente el proceso es sencillo; Tras haber probado varias veces el valor de potencia para los motores necesario para realizar la rotación necesaria, usando la función OnFwd le damos a ambos motores esta potencia. Estableciendo el tiempo deseado en milisegundos, hacemos un bucle que vaya desde el tiempo inicial de la función CurrentTick hasta el tiempo que establecimos al comienzo. En este bucle indicamos que se lea e imprima por pantalla los valores de rotación de ambos motores. Una vez finalice el tiempo que se indicó en un principio se detienen los motores mediante la función Off seleccionando ambos motores y se indica que el programa ha finalizado.

## EJERCICIO 1B

En este ejercicio, lo hemos realizado de dos formas, una de ellas hacía que el robot fuese en línea recta empezando con una velocidad determinada y después de 5 segundos pararía, vería cuantos grados han girado sus ruedas, y comparando con su objetivo de hacer  $860^\circ$ , vería si le faltaba más o menos potencia, así en la siguiente iteración usaría la potencia que cree adecuada según lo que ha visto en la anterior iteración, y después de 5 segundos,

volvería a comparar lo girado con el objetivo que se le ha dado, así repetiría hasta 5 veces acercándose a una potencia que le sea suficiente para hacer los 860° de una sola vez.

Todo sería personalizable, tanto las rotaciones objetivo, como iteraciones y el tiempo entre iteraciones, mostrando al final en una tabla lo que se ha conseguido en cada iteración.

En cambio en el código que hemos enviado, se busca que el robot se pare en 860° con muy poco error. Empieza intentando llegar con una potencia determinada a ese objetivo, después de 5 segundos parará, verá cuantos grados le falta o le sobra para llegar a 860°, y ajustará la potencia a la que cree necesaria para llegar. Volverá a intentar llegar a ese objetivo con esa nueva potencia, y una vez pasado 5 segundos, comprobará cuanto le queda para llegar y adaptará la potencia de nuevo, así en 3 iteraciones. Todo esto se puede personalizar y por lo tanto se puede dar más iteraciones, cambiar el tiempo, el objetivo y la potencia inicial. Cuando termina de ejecutar el código, se mostrará en una tabla la potencia usada en cada iteración, la rotación actual del robot, la rotación objetivo y la diferencia entre estas rotaciones.

Para saber cuanta potencia debe hacer para recorrer la diferencia de grados para llegar al objetivo, se usa una fórmula en la cual al darle los grados que le falta por recorrer, lo adapta a grados por segundo y luego se convierte en potencia necesaria para hacer esos grados por segundo. Esta fórmula es de una simple recta ( $y = m \cdot x + a$ ) la cual se ha calculado gracias a comprobar cuantos grados recorría en 1 segundo con 90 de potencia, cuantos con 45 y cuantos con 5.

## **EJERCICIO 2**

Para el seguimiento de líneas, el objetivo es que dada una luminosidad R, el robot recorriese la parte de la elipse que tuviese esa luminosidad, para ello se situará a este sobre la parte de la elipse en la cual capte una luminosidad bastante cercana o igual a la objetivo, así podrá encontrar el camino que debe recorrer.

La programación es simple, si ve que esta más claro del objetivo deseado, entonces girará hacia la izquierda respecto a su eje Z, en cambio si es más oscuro, girará hacia su derecha, pero si la luminosidad está en el rango [R-U,

$R+U]$ , entonces girará recto, siendo  $U$  el umbral permitido. Si no encuentra una luminosidad dentro del rango, entonces se quedará dando vueltas hasta que encuentre un valor que si esté dentro del rango.

Toda la luminosidad leída en cada momento, se escribe en un archivo llamado Datos, el cual contiene el tiempo y la luminosidad de ese momento.

Si se varía la potencia del robot,  $P$ , entonces se podrá comprobar que a mayor potencia, más velocidad tendrá y le será más difícil al robot ser preciso, en cambio a menor potencia, se verá que recorre mucho mejor el camino sin dar bandazos como pasa a grandes potencias, y por lo tanto tomará con más cuidado las curvas y no le hará falta corregir tanto la dirección como le pasa cuando se le da demasiada potencia, eso si, no se le puede dar muy poca potencia al robot, ya que en ciertas potencias bajas, no podrá ni moverse del sitio.