

## **Grupo:**

Antonio David Ponce Martínez  
Sergio González Muriel

## **Objetivo de la práctica:**

Esta práctica se dividía en dos partes:

1. Experimentación con el control en bucle abierto y cerrado
2. Implementación de un seguimiento básico de líneas

## **Parte 1: Control en bucle abierto y cerrado:**

### **Propiedades:**

#### **a) Control en bucle abierto:**

- Es aquél en el que la salida del control no se compara con la entrada.
- Es mucho más impreciso que el de bucle cerrado.
- Hay que definir más parámetros, ya que necesita ser calibrado.

#### **b) Control en bucle cerrado:**

- Es aquél en el que la salida del control se compara con la entrada.
- Es muy preciso.
- Utilizado en modelos autorregulables.

### **Problemas encontrados:**

Definir la potencia correcta en el control en bucle abierto:

- Un problema principal del control en bucle abierto es que nosotros teníamos que ir probando diferentes potencias hasta dar con una que nos diese el número de revoluciones que necesitábamos.
- Esto lo resolvimos probando hasta dar que con un valor de potencia de 50, realizaba 860 revoluciones más o menos. (Apoyado en la mesa)

Mostrar el número de revoluciones tras 2 segundos de ejecución:

- En un principio no sabíamos cómo afrontar este problema, debido a que a los 2 segundos teníamos que apagar los motores pero no la ejecución, ya que durante los siguientes 3 segundos teníamos que mostrar por el display del robot el número de revoluciones que habían dado sus ruedas.
- Así, conseguimos solventarlo añadiendo el Clock como un timer a la función de Matlab y, cuando el tiempo es de 2 segundos, ponemos la potencia de los

motores a 0 y devolvemos un entero "has\_finished" que funciona como booleano para un "if block".

- La ejecución se para cuando el reloj de simulink indica 5 segundos.

### **Resultados obtenidos:**

La diferencia entre los dos tipos de control es su precisión. Debido a que en el control en bucle abierto tenemos que introducir nosotros mismos la potencia a suministrar a las ruedas y eso se realiza mediante mediciones previas, produce error en las medidas en otras situaciones distintas a las comprobadas. Así pues, si el robot andaba por la mesa, producía un resultado cercano al buscado (860), sin embargo, si lo cogíamos, las ruedas giraban más de 860 vueltas, debido a que, en el aire, no contaban con la resistencia que producen las ruedas con la mesa.

Al contrario, con el control en bucle cerrado, siempre nos proporcionaba un resultado cercano al buscado, debido a que la potencia suministrada a las ruedas se autorregula dependiendo de las vueltas que ya hayan dado las ruedas.

## **Parte 2: Seguimiento básico de líneas:**

### **Problemas encontrados:**

Valores negativos en simulink:

- En las instrucciones proporcionadas, nos explicaban que teníamos que utilizar un error para calcular la potencia suministrada por las ruedas.
- Este error podía estar entre unos umbrales (R-U y R+U), por encima o por debajo.
- Sin embargo, al calcular el error, si salía negativo e intentábamos sacarlo fuera del bloque de la función en Simulink, Simulink lo consideraba como 0.
- Así pues, lo solucionamos desglosando el error y calculando su valor indirectamente con las condicionales (sentencias if), no produciendo ningún número negativo.

Array de tiempos e intensidades:

- En la práctica nos explicaban que teníamos que guardar un array de tiempos e intensidades para, posteriormente, imprimirlos en un fichero. Esto supuso un problema al principio porque estas variables tienen que ser persistentes y, un array, si es persistente, no puede cambiar su tamaño en cada iteración.

- Esto lo hemos afrontado creándose como un array de 0's de tamaño el número de segundos que se está ejecutando la función (en este caso 5) por 1000, que es la frecuencia del reloj en simulink.

### Imprimir al fichero:

- En la última sesión de laboratorio encontramos dificultades a la hora de imprimir al fichero debido a los array persistentes y a los tipos de dichos arrays. Además no teníamos tiempo de arreglarlo en la clase.
- Así pues, ejecutamos las pruebas sin imprimir en el fichero y corregimos el código posteriormente a la clase.

### **Resultados obtenidos:**

A pesar de no tener los ficheros de los datos, debido a observaciones hemos visto que para valores bajos de  $P$ , los valores de las intensidades varían muy progresivamente o incluso ni varían, ya que el robot se mueve muy lento. Sin embargo, cuanto más alto es el valor de  $P$ , las intensidades varían de forma más brusca, debido a que el robot se mueve muy rápido y al sensor no le da tiempo de captar todas y cada una de las intensidades por las que pasa el robot.