Prácticas de Robótica

Práctica 2 - "Slalom".

Generación de trayectorias y control de movimiento

Objetivos

1. Implementar funcionalidad para **gestionar velocidades** (lineal y angular) del robot.
2. Implementar funcionalidad para estimar en todo momento la **odometria del robot**
3. Utilizando lo anterior, generar una **trayectoria determinada** con el robot.

Antes de nada:

- El robot tiene que estar montado para poder hacer esta práctica.
- En moodle encontrareis los ficheros base para esta práctica con los requisitos de las funciones a implementar.
  Para evaluar las entregas de las prácticas los profesores asumen que la definición de las funciones requeridas es como en dichos ficheros (mismo nombre y número de parámetros). Si se cambia, explicar claramente en un README en la entrega.
- En la plantilla de la práctica hay código de ejemplo que tendreis que mirar y decidir que quereis usar para **gestionar variables compartidas entre tareas/procesos**. Por defecto se usa la biblioteca `multiprocessing` de python. Podeis cambiarlo y usar `Thread` u otras si preferis (pero fijaros en que multi-process aprovecha varios cores, y Thread no).

Descripción

1. Velocidades del robot:

Mediante el análisis del mecanismo de tracción del robot, desarrollar estas dos funciones (y las auxiliares que se consideren necesarias) que nos permitan asignar o leer las velocidades (lineal **v** y angular **w**) que lleva el robot:

`setSpeed(v,w)` → dentro de la función debemos asignar a los motores correspondientes la potencia o velocidad necesaria para alcanzar esta velocidad.
`[v,w] = readSpeed()` → la función debe devolver los valores actuales de v y w del robot.

2. Odometría:

El robot debe mantener una variable "interna" actualizada constantemente, con la posición y orientación actual (x,y,θ) respecto al punto donde arrancó, para poder consultarla en cualquier momento. Esta tarea se lanza en un **proceso en paralelo**, como se muestra en la plantilla. En clase se ha explicado el funcionamiento de las funciones utilizadas para ello, no es necesario modificar nada de este aspecto.
Desarrollar estas dos funciones y las auxiliares que se consideren necesarias:

`updateOdometry()` → actualiza los valores de odometría periódicamente

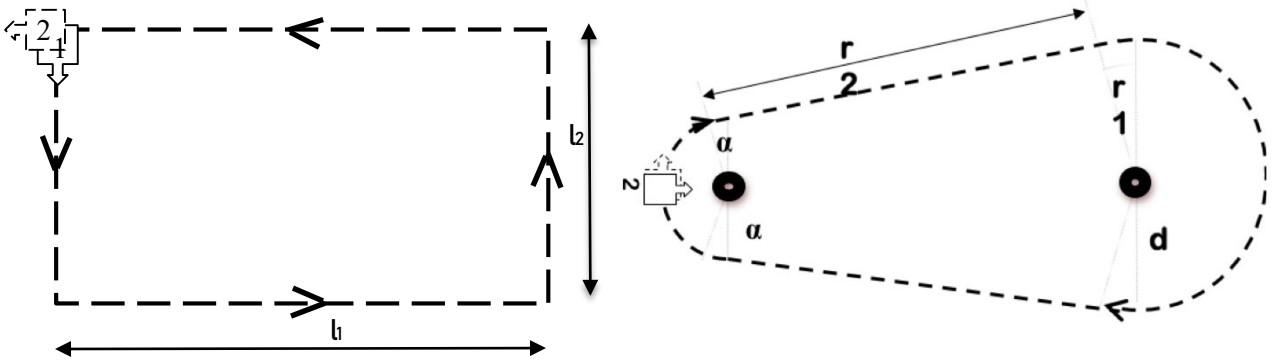
[x, y, th] = `readOdometry()` → devuelve los valores actuales de la odometría

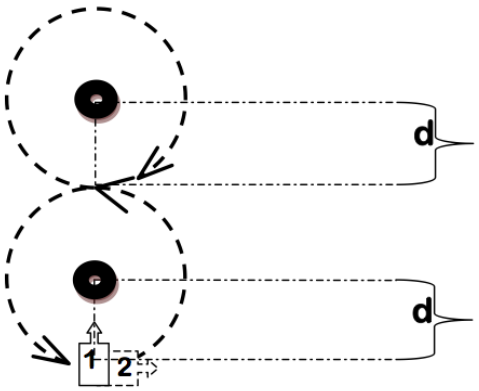
VERIFICAR:

- si el robot **solo tiene velocidad lineal** con el setSpeed (v, 0), va recto y la odometría solo modifica sus componentes **x, y;**
- si **solo tiene velocidad angular**, setSpeed (0,w) gira sobre sí mismo y en los valores de la odometría prácticamente solo cambian en la th.
- La odometría estima bien cuando el robot gira 90º sobre sí mismo, o cuando avanza una baldosa recto (40 mm approx.)

3. Generar trayectorias:

Utilizando lo anterior, hacer que el robot realice una trayectoria predeterminada. **Obligatorio realizar todas las trayectorias**. Asignar radios **d** y **α** que querais, asi como las diferentes distancias. **Guardar en un fichero** de texto el valor de odometría cada poco tiempo a modo de "LOG" para luego visualizarlo. **El objetivo NO es tener error 0!!** Sino **medir el error** que tiene la odometría realizando trayectorias sencillas o más complejas.





Trabajo PREVIO (se evaluará al entrar a la práctica en un cuestionario)

- Preparar (papel o código) **pseudocódigo/esquema en python (**con las expresiones matemáticas que necesitais) para implementar el **`setSpeed(v,w):`**

- Preparar **pseudocódigo/esquema en python** para **`updateOdometry():`**

- Estimar las distintas **partes y v,w necesarias** para realizar las trayectorias:

Descripción de trayectoria 1	v	w	Tiempo de ejecución
Descripción de trayectoria 2	**v**	**w**	**Tiempo de ejecución**
Descripción de trayectoria 3 (el ocho)	**v**	**w**	**Tiempo de ejecución**

Evaluación

- **Entrega del código y LOG+Plot de odometría de trayectoria generada, en moodle, el día establecido para la entrega de la tarea.** Se valorará:
  - Claridad y legibilidad del código. Al menos se recomienda incluir un "docstring" en cada una de las funciones principales.
  - Funcionalidad correcta de las funciones requeridas
- **Demo** a los profesores de las tres tareas realizadas. Bien durante la sesión si se terminan, o al empezar la sesión de la práctica siguiente.