

Percepción y Control
PRÁCTICA FINAL

**Diseño de guiado de un robot
móvil explorador**

Ingeniería de Computadores

Pedro Barquín Ayuso
Miguel Ballesteros García

Índice

1.-Estrategia de exploración.....	3
1.1-Descripcion de la estrategia elegida.....	3
2.-Percepcion y control.....	4
2.1-Navegacion sin error de odometría.....	4
2.1.1-Sonar.....	4
2.1.2-Laser.....	4
2.1.3-Giro.....	4
2.1.4-Avance.....	5
2.2-Navegacion con error de odometría.....	5
2.2.1-Avance.....	5
3.-Mapeado completo.....	6

1.-Estrategia de exploración

Como estrategia de exploración hemos utilizado el algoritmo de seguir la pared derecha. Sabemos que no es el algoritmo para desarrollar robots de este tipo y dependiendo del mapa puede no llegar a completarlo correctamente, por lo que hemos desarrollado un algoritmo que rellena las casillas sin visitar en función de las adyacentes conocidas.

Contamos con una matriz de 11x11 donde vamos a ubicar las casillas visitadas para la posterior creación del mapa. La matriz debe ser del tamaño elegido para contener el mapa independientemente del lugar de inicio del recorrido.

Cada casilla de la matriz es una estructura compuesta por:

- casillaNueva: nos indica si la casilla ha sido visitada anteriormente o no
- marca: nos indica si es el inicio del recorrido o un punto del camino
- veces: es un contador que nos indica el número de veces que se ha visitado cierta casilla
- formato: nos indica el numero del tipo de casilla en la que estamos en función de las pareces que lo rodea.

1.1-Descripción de la estrategia elegida

Antes de empezar inicializamos la matriz para evitar fallos a la hora de recorrerla. Antes de cada iteración se realiza una comprobación en la que miramos si no hay ninguna casilla visitada más de cuatro veces (lo que supondría que hemos completado el recorrido) o si el numero de casilla que faltan por visitar es mayor que 0 dado que sabemos que el número de casillas máximo es 25.

Comprobamos mediante los sensores y la odometría en qué situación estamos (formato de casilla en la que nos encontramos, ubicación). Con estos datos nos disponemos a rellenar la matriz principal siguiendo los siguientes pasos:

- Redondeamos la ubicación al entero más cercano para poder ubicarnos en la matriz
- Comprobamos si es la primera vez que se visita dicha casilla, si no ha sido visitada entonces rellenamos la casilla con la información obtenida y aumentamos su contador de veces visitada y decrementamos el numero de casillas totales a visitar. En caso contrario solo aumentamos el contador de veces visitas dado que la casilla ya esta rellena.

Una vez rellena la casilla efectuamos la acción necesaria para salir de ella en función del tipo de formato de la casilla en la que nos encontremos. La elección de los movimientos se ha elegido en base al algoritmo de seguir la pared derecha y repetimos todo el proceso hasta completar el recorrido.

2.-Percepcion y control

2.1-Navegacion sin error de odometría

2.1.1-Sonar

Hemos utilizado este sensor para comprobar la situación de las paredes que rodean al robot en cada una de las casillas. Para la detección de las paredes medimos la distancia frontal, lateral y trasera que nos da el sonar y suponiendo que el robot esta en el centro de la casilla podemos estimar gracias a la medida y al coeficiente de pearson si tenemos una pared y su grado de confianza. Sabiendo las paredes que rodean al robot podemos saber el formato de la casilla en la que nos encontramos y actuar acorde.

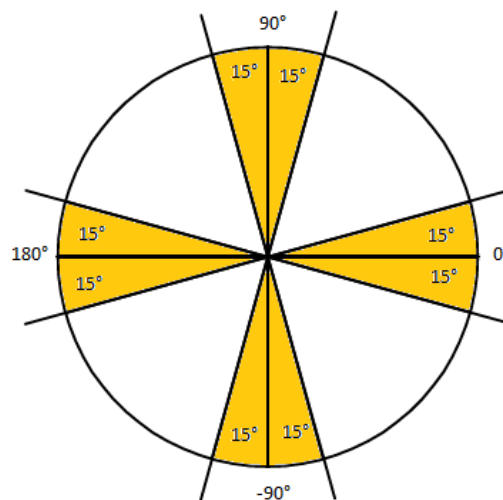
2.1.2-Laser

Este sensor lo utilizamos como comprobación para saber la distancia entre el robot y la pared frontal según vamos avanzando. Esta información se utilizara en el control del avance del robot.

2.1.3-Giro

Esta operación se realiza para cambiar la orientación del robot antes de realizar otro movimiento. Efectuamos esta acción siguiendo los siguientes pasos:

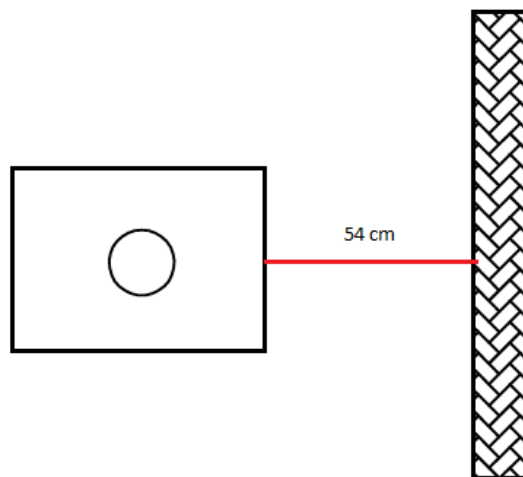
- En función de nuestro ángulo actual y el giro que se nos imponga obtenemos la nueva orientación (0, 90, 180,-90) aproximada para evitar errores
- Dentro de cada orientación nos podemos mover a dos velocidades diferentes. Velocidad rápida cuando estamos a más de 15º de la orientación deseada. Velocidad lenta cuando estamos a menos de 15º de la orientación deseada para hacer que el giro sea lo más preciso posible.



2.1.4-Avance

Para el avance nos hemos basado en la práctica del control P, corriendo de esta manera el avance y la orientación del robot al moverse de una casilla otra. Efectuamos esta acción siguiendo los siguientes pasos:

- Obtenemos la ubicación del robot
- A partir de la posición del robot y el avance a realizar obtenemos la posición final a la que queremos llegar.
- Aproximamos la ubicación destino
- Mientras que estemos a una distancia mayor de 0.12 respecto a nuestro objetivo y el laser nos dé una distancia mayor a 0.54 de la pared, realizamos la operación de avanzar. Las operaciones a realizar durante el avance serán controladas en todo momento por un control P



2.2-Navegacion con error de odometría

En nuestro caso solo controlamos el error lineal de odometría sin tener en cuenta el error angular. En este caso los apartados laser, sonar y giro serian idénticos, modificando únicamente el apartado del avance.

2.2.1-Avance

Para este caso se ha suprimido el error angular del control P y se ha añadido la calibración de la odometría mediante la función `set_odom`. Esta función se realiza al terminar cada operación de avanzar y los parámetros que pasamos son los estimados para cada nuevo avance.

3.-Mapeado completo

Una vez que el robot estima que ha terminado el recorrido, representamos:

- Una leyenda del mapa en la que se aprecio el formato que tiene cada número de casilla.

```
El mapa esta representado mediante la informacion de la tabla en la que el simbolo + representa un hueco libre y el simbolo - o | representa una pared, se ha
de tener en cuenta que el numero de la casilla del mapa corresponede con los dibujos
+++++ +---+ +---+ +---+
+ + + + + | + +
+ + + + + | + +
+++++0 +---+1 +---+2 +---+3

+++++ +---+ +---+ +---+
| + + | + + | +
| + + | + + | +
+++++4 +---+5 +---+6 +---+7

+++++ +---+ +---+ +---+
+ | | | | + + +
+ | | | | + + +
+---+8 +---+9 +---+10 +---+11

+++++ +---+ +---+ +---+
| | | + | | | |
| | | + | | | |
+---+12 +---+13 +---+14 +---+15
```

- Mostramos un dibujo aproximado de las casillas visitadas con su número correspondiente.

```
Representacione de las casillas del mapa en numero

13  5  7  6  11
7   8  10 1  5
10  5      9  12
14  4  1  3  5
10  8  10 11  12
```

-Mostramos un esquema de las casillas del mapa por las que ha pasado el robot y la posición inicial del robot.

Representacion de los lugares visitados inicio (X) y resto (*)

```

      *  *  *  *  *
      *  *  *  *  *
      *  *      *  *
      X  *  *  *  *
      *  *  *  *  *
  
```

-Finalmente, mediante el algoritmo rellenarBlanco mostramos el mapeado final completo.

Representacion de las casillas del mapa en numero con los huecos rellenos

```

      13  5  7  6  11
      7  8  10  1  5
      10  5  15  9  12
      14  4  1  3  5
      10  8  10  11  12
  
```

El algoritmo rellenarBlanco busca casilla sin visitar que se encuentren alrededor de casillas conocidas y comprobando cuatro pares posibles de casillas conocidas se realiza una estimación de la casilla que nos falta.

