

## Introducción a la Robótica Inteligente:

### Trabajo Voluntario 1

**Alumno: Andrés Ripoll Cabrera.**

- Implementar en el simulador IRSIM las ecuaciones cinemáticas que permiten obtener la posición y orientación relativa del robot a partir de los datos suministrados por el encoder:

Para el desarrollo de las ecuaciones cinemáticas he empleado odometría, buscando información en un documento del departamento de electrónica de la Universidad de Alcalá. En este, se explican tanto las ecuaciones para obtener las posiciones x e y del robot como la orientación. Para obtener las posiciones x e y, se suma la posición anterior a la suma entre las distancias que avanzan las ruedas(encoders) dividido entre dos y multiplicado, en caso de la x por el cos de la orientación (el de el mismo instante de tiempo que queremos calcular), y en el caso de la y el sin de la orientación. La orientación se calcula como la orientación en el instante anterior más la distancia que avanza la rueda derecha menos la distancia que avanza la rueda izquierda entre la distancia entre ruedas(*fWheelsDistance*).

$$Pos.x = Pos.x_{Inicial} + \frac{(encoder[0] + encoder[1])}{2} * \cos(m\_fOrientation)$$

$$Pos.y = Pos.y_{Inicial} + \frac{(encoder[0] + encoder[1])}{2} * \sin(m\_fOrientation)$$

$$m\_fOrientation = m\_fOrientation_{Inicial} + \frac{(encoder[1] - encoder[0])}{fWheelsDistance}$$

Hay que destacar que estas ecuaciones se realizan en base a la posición y orientación anterior del robot, por lo que para que funcionen correctamente se coloca el robot en la posición (0,0,0) inicialmente.

Una vez realizadas las ecuaciones realicé un par de pruebas. Si el robot se movía en dirección recta a izquierda o derecha no daba lugar a error. Las complicaciones empezaron cuando simplemente el robot daba vueltas sobre sí mismo, en sentido antihorario. Aquí, el problema era con la orientación, ya que no dejaba de aumentar su valor y solo era correcto para la primera vuelta sobre sí mismo. Para solventar esto, se me ocurrió añadir un if, que decía que si la orientación era mayor que  $2 * \pi$  ( $360^\circ$ ), debía restar esta misma cantidad a la orientación. De esta manera volvería a estar dentro del rango de 0- $360^\circ$ . El segundo problema fue que el robot no actualizaba correctamente la orientación cuando giraba en el sentido

de las agujas del reloj. Aquí utilizaba radianes en negativo, por lo que añadí otro if, que para valores de orientación negativos, se le sumaba  $2 * \pi$  ( $360^\circ$ ), es decir, una vuelta completa para dar su valor en positivo.

Las últimas pruebas que realicé fueron que el robot diese vueltas sobre el centro (0,0). Al haber arreglado la orientación para el caso en el que el robot rotaba, aquí no da problemas, pero las posiciones x e y dan un pequeño error, de milésimas, cuando el robot arranca, pero que se mantiene estable cuando describe la trayectoria circular. Este error no he conseguido corregirlo ya que desconozco exactamente su origen. Tal vez sea debido a el error que proporcionan las funciones sin y cos, pero de todas formas no ha conseguido averiguarlo.

#### Bibliografía:

[http://www.roboticaeducativa.org/file.php/77/IDMRM1213\\_Actividades\\_5\\_1\\_Mo\\_nitorizandoMovimiento.pdf](http://www.roboticaeducativa.org/file.php/77/IDMRM1213_Actividades_5_1_Mo_nitorizandoMovimiento.pdf)